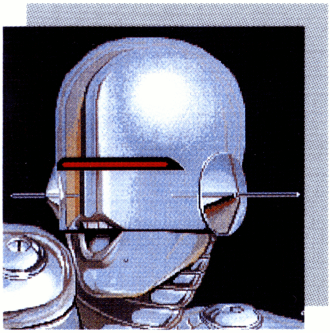
AMIGA in der Tasche

Ralf Leithaus (Hrsg.)

Public Domain Schatztruhe

Wegweiser durch freie AMIGA-Software



Verlag technicSupport

Public Domain Schatztruhe

AMIGA in der Tasche Band 1

Ralf Leithaus (Hrsg.)

AMIGA in der Tasche Band 1

Public Domain Schatztruhe

Wegweiser durch freie AMIGA-Software

Verlag technicSupport

Leithaus, Ralf (Hrsg.)

Amiga Public Domain Schatztruhe

Wegweiser durch freie Software

Berlin (West): technicSupport 1990

ISBN 3-926847-22-0

© Copyright 1990 by technicSupport GmbH

Das Werk einschließlich aller Teile ist urheberrechtlich geschützt. Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe, Vervielfältigungen anderer Art, Übersetzungen, Mikroverfilmung und der Speicherung/Bearbeitung in elektronischen Systemen. Die gewerbliche Nutzung der dargestellten Beispiele und Programme ist nicht zulässig. Jede Verwertung bedarf der schriftlichen Genehmigung des Verlages oder ggf. der Copyright-Inhaber.

Bei der Zusammenstellung von Texten, Abbildungen und Beispielen wurde mit großer Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren übernehmen weder eine juristische Verantwortung noch eine andere Haftung. Der Verlag übernimmt auch keine Gewähr dafür, daß Programme erhältlich sind und tatsächlich funktionieren.

Der Verlag macht darauf aufmerksam, daß die genannten Markennamen und Produktbezeichnungen in der Regel marken-, patent- oder warenzeichenrechtlichem Schutz unterliegen. Das vorliegende Werk verwendet die Markennamen und Produktbezeichnungen, ohne daß dies jedes Mal gekennzeichnet wird. Eine Verwendung der geschützten Markennamen bedarf der schriftlichen Genehmigung der Schutzrechtinhaber.

ISBN 3-926847-22-0

Computersatz:	Ralf Leithaus
Satz-Software:	AmigaTEX 2.0
Satzbelichtung:	Rößler & Partner, Konstanz
Bildbearbeitung:	Holger Stück, Ralf Leithaus
Titelgestaltung:	Felix Schröder
Druck:	Felgentreff & Goebel, Berlin

Inhalt

Vorwort 7

Einführung 11

Anwendungsbeispiele 23

Workbench 1.3 für Individualisten 25
Die PD-Story 43

Programmbeschreibungen 57

AmiCron — Startet Programme zu bestimmten Zeiten 59
ARP 1.3 — Doping für die Workbench 62
AsciiTable — Anzeige der Tastencodes 74
AssignDev — Umbenennen physikalischer Geräte 76
CardFile — Ein elektronischer Karteikasten 79
Chess — Schachprogramm in zwei Entwicklungsstadien 88
Conman — Editierhilfe für CLI/Shell 96
DFC2 — Kopiert und formatiert Disketten 103
DiskMan — Konfigurierbarer Disk-Manager 107
Dme — Ein Editor zum Maßschneidern 116
ExplodingWindows — Effektvolles Öffnen von Fenstern 134
Flaschbier — Das Werner-Spiel 136
FontCon — Wandelt Zeichensätze in Druckerfonts 140
Freud — Ein interessanter Persönlichkeitstest 145
FuncKey — Belegung der Funktionstasten 153
Gagprogramme — Die besten Programmierspäße 159
Hack — Graphisches Abenteuerspiel 164
Hexalator — Umrechnen verschiedener Zahlenformate 176
Hyperbase — Menügesteuerte Dateiverwaltung 178
IconType — Ändert den Typ eines Icons 190
KeyClick — Akustisches Feedback bei Tastatureingaben 192
less — Textdateien unter der Lupe 194
MachII — Viele Utilities in einem Programm 201
Mackie — Bildschirmschoner mit Extras 209
MandelVroom — Apfelmännchen-Generator 214

MaxWindow — Bringt CLI-Fenster auf PAL-Größe 243
Moose — Das Ikea-Imperium schlägt zurück 244
MouseZoom — Ein etwas anderer Mausbeschleuniger 250
Movie — Animationen mit weniger Speicherbedarf 251
NECquik — Druckerkontrolle von der Workbench 258
NewZap — Dateiorientierter Diskettenmonitor 261
Paranoid — Breakout-Spiel mit Level-Editor 266
PLAY — Jukebox für SONIX-Musikstücke 269
PrintPop — Druckerkommandos von der Workbench 275
PointerAnimator — Der Mauszeiger wird animiert 278
QView — Schneller Textbetrachter 285
Ren — Auslagern von Zeichensatzdateien 287
RunBack — Starten von Hintergrundtasks 292
SafeBoot — Archiv für Bootblöcke 295
SetPrefs — Lädt beliebige System-Configuration 298
ShoWiz — Slideshow-Programm 301
startle — Von der Vision eines Hackers 309
StoneAge — PD-Variante zu Emerald Mine 311
TurboBackUp — Schnelles Diskettenkopieren 315
VC/AnalytiCalc — Tabellenkalkulation 318
Vdraw — Malprogramm für verschiedene Auflösungen 335
VirusX — Automatischer Virusdetektor 350
WBLace — Schaltet Workbench auf Interlace 354
WBLander — Kleines Spiel auf der Workbench 355
XIcon — Startet DOS-Kommandos von der Workbench 356
ZapIcon — Generiert Icons aus IFF-Bild 365

Liste aller Fishdisks 367

Autorenverzeichnis 403

Register 405

© 1990 technicSupport

Vorwort

Herzlich willkommen im Paradies der frei kopierbaren Software für Amiga. „Public Domain Software“, so lautet die Zauberformel, mit der Amiga-Profis ihren Computer auf Trab bringen, für mehr Abwechslung sowie komfortableres und erfolgreicheres Arbeiten sorgen, ohne viel Geld hinblättern zu müssen. Aber auch für frischgebackene Amiga-Besitzer bietet PD-Software ideale Möglichkeiten für eine praxisorientierte Einarbeitung. Das riesige Spektrum der Public Domain Software reicht von kleineren Hilfsprogrammen, die so manche Schwierigkeit aus dem Weg räumen, über Programme und Demos aus den Bereichen Grafik und Musik, bis hin zu hochkomplexen Programmen aus Wissenschaft, Büroanwendung und Programmierung. Nicht zu vergessen die zahlreichen Spiele, die dafür sorgen, daß neben der ernsthaften Anwendung auch die Unterhaltung nicht zu kurz kommt.

Der Berliner Verlag technicSupport beschäftigt sich bereits seit den ersten Tagen des Amiga mit der frei kopierbaren Software für dieses Computer-System. Aus dieser Beschäftigung entstand dann im Frühjahr 1988 DAS GROSSE AMIGA PUBLIC DOMAIN BUCH, in dem bereits rd. 50 der damals besten PD-Programme vorgestellt wurden. Die Flut der guten PD-Programme für Amiga ist seither jedoch beständig angestiegen, so daß aus dem Einzelband inzwischen eine vierbändige PD-Bibliothek geworden ist. 18 Autoren haben unter der Leitung von drei Herausgebern ein Standardwerk zur Amiga-PD mit fast 1.700 Seiten Programmbeschreibungen und nützlichen Informationen geschaffen. Alle 240 inzwischen beschriebenen Programme liegen auch auf einer eigenen PD-Reihe vor und werden ebenfalls vom Verlag technicSupport geliefert.

Als preiswertere Alternative zu dieser großen Bibliothek und besonders für PD-Einsteiger ist das vorliegende Taschenbuch konzipiert worden. Zu einem günstigen Preis präsentiert dieses Buch einen repräsentativen Querschnitt von Programmbeschreibungen aus

der PD-Bibliothek und damit einen Querschnitt durch die Amiga-PD-Software schlechthin. Neben 51 praxisnahen Programmbeschreibungen aus nahezu allen Anwendungsgebieten enthält es eine komplette Liste der berühmtesten und wichtigsten PD-Serie von FRED FISH, die zur Zeit der Herausgabe dieses Taschenbuchs 282 Disketten mit mehr als 1600 Programmen umfaßt. Auch zum Taschenbuch ist beim Verlag ein Paket mit drei Disketten und 32 ausgewählten Programmen erhältlich.

Die meisten Programmbeschreibungen dieses Buches gehen somit auf die vier Bände der PD-Bibliothek zurück, sind jedoch in den Fällen, wo es Weiterentwicklungen von Programmen gegeben hat, auf den neuesten Stand gebracht worden. Dadurch erhält der Leser die größtmögliche Gewähr, mit den jeweils neuesten Programmversionen zu arbeiten.

Daß ein preiswertes Buch nicht automatisch eine Billigproduktion sein muß, wird beim Durchblättern der folgenden Seiten klar. Wie alle anderen technicSupport-Bücher ist auch dieses Buch komplett auf Amiga erstellt worden. Verwendet wurde für diese Aufgabe das Schriftsatzprogramm AmigaTEX, das in der neuen Version 2.0 nicht nur den gesamten Umbruch steuert, sondern auch die Einbindung von IFF-Grafiken erlaubt. Dadurch wurde es möglich, die Seiten einschließlich aller Bilder auf Amiga zu gestalten und sie anschließend auf einem Satzbelichter mit 1200 dpi Auflösung (das ist die vierfache Auflösung eines herkömmlichen Laserdruckers) auszugeben. Besonders danken möchten wir Herrn JOCHEN HÄBERLE und der Fa. RÖSSLER & PARTNER für ihre Initiative und ihr Engagement, AmigaTEX für einen Satzbelichter einzurichten, aber auch TOMAS ROKICKI, dem Programmierer von AmigaTEX, der dieses Programm regelmäßig um neue und nützliche Funktionen erweitert.

Ein herzliches Dankeschön geht aber auch an alle Autoren, die durch Einsatz, Mitarbeit und nicht zuletzt durch hervorragende Programmbeschreibungen das Gesicht der PD-Bücher von technicSupport geprägt und ihnen zu ihrer führenden Stellung verholfen haben: WERNER GÖRTZ, HOLGER LUBITZ, STEFAN RAM, OTTMAR RÖHRIG, MICHAEL SAUER, AXEL SCHMIDT, KATJA THIELBEER und KARL-WILHELM ZERULLA.

 © 1990 technicSupport

Besonderen Dank schuldet der Verlag jedoch Ralf Leithaus. Als Herausgeber von drei Bänden der PD-Bibliothek und dieses Taschenbuches hat er mit seinem Einsatz und seiner Kenntnis über die freie Software wesentlich zur Verbreitung der PD in Deutschland beigetragen. Daß ein Herausgeber sich nicht nur auf die inhaltliche Arbeit konzentriert, sondern auch den Schriftsatz seiner Bücher selber in die Hand nimmt, ist nicht nur ungewöhnlich. Es zeigt sicher auch eine besondere Verbundenheit zum Amiga, Schriftsatz in dieser Qualität auf ebendiesem System mit AmigaTEX zu erstellen. Dafür herzlichen Dank an den Herausgeber.

Wir wünschen allen Lesern, die über dieses Taschenbuch zum ersten Mal mit PD-Software in Kontakt kommen, daß sie den Wert dieser oft noch unterschätzten Software-Gattung erkennen. Und übrigens: Die Beschäftigung mit dem Amiga soll auch Freude bereiten. Viel Spaß also mit diesem Buch und „The Best of Amiga-PD“!

Jens Hertwig
Verlagsleitung

© 1990 technicSupport

Einführung

„Public Domain“ — mit diesem Stichwort konnten bis vor etwa zwei Jahren nur wenige Insider etwas anfangen. Heutzutage wird man dagegen kaum einen Amiga-Anwender treffen, der nicht weiß, daß mit diesem Begriff eine bestimmte Software-Gattung bezeichnet wird: Software, die frei kopierbar ist, die man also für sehr wenig Geld erwerben und besitzen darf. Auch der Erwerb dieser Software bereitet heutzutage keine Schwierigkeiten mehr. Man kann sich die PD-Sammlung eines Freundes kopieren, man kann PD-Programme per DFÜ von einer Mailbox herunterladen, man kann die bekannten Diskettenreihen bei Versandanbietern bestellen, und manchmal wird man sogar beim Amiga-Händler um die Ecke fündig.

PD-Software ist aber nicht gleich PD-Software. Während man PD-Software in beliebigen Mengen sammeln darf, ohne rechtliche Schwierigkeiten zu befürchten, sind bei Benutzung, aber auch bei Weitergabe der meisten Programme verschiedene Richtlinien zu beachten, die in der Regel den programmbegleitenden Dokumentationsdateien zu entnehmen sind.

Die Bezeichnung „Public Domain“ hat sich im deutschen Sprachgebrauch als Gegenbegriff zur „kommerziell genutzten Software“ eingebürgert. Dieser Sprachgebrauch täuscht oft darüber hinweg, daß nur die wenigsten Programme, die auf Public-Domain-Disketten in Umlauf sind, „echte“ Public-Domain-Software sind. Public-Domain-Programme im eigentlichen Sinne des Wortes sind nur solche Programme, bei denen der Autor auf sämtliche Rechte verzichtet. Diese Programme dürfen Sie in beliebiger Weise verwenden und sogar gewerblich nutzen. Mittlerweile sind die meisten Programmierer zu der Einsicht gelangt, daß es nicht im Sinne des Erfinders sei, wenn andere die Früchte ihrer geistigen Arbeit ernten. Sie verbinden daher die Nutzung ihrer Progamme mit der Einhaltung bestimmter Auflagen. Diese Programme sind mit einem Copyright versehen, welches beispielsweise die gewerbliche Nutzung untersagt oder die Ver-

wendung innerhalb eigener Programme an die Beibehaltung dieses Copyright-Hinweises bindet. Welche genauen Auflagen im Einzelfall einzuhalten sind, müssen Sie den zu den Programmen gehörenden Textdateien entnehmen.

Ein besonders interessantes Konzept innerhalb der Public Domain ist das der *Shareware*. Der Autor eines Shareware-Programmes gestattet Ihnen die private Nutzung, wünscht sich jedoch eine Anerkennung in Form eines kleineren Geldbetrages, falls Ihnen das Programm so zusagt, daß Sie es häufiger benutzen. Oft erhalten Sie für dieses Geld sogar noch ein Handbuch oder einen Update-Service, zudem unterstützen Sie den Autor bei der Entwicklung neuer Programme, deren Nutznießer Sie wiederum selbst sein können. Da es de facto keine Kontrollmöglichkeiten für die Verwendung von Shareware gibt, hängt es von uns PD-Anwendern ab, ob sich diese Idee durchsetzt. Wenn man statt eines teuren Programmes ein Shareware-Programm benutzt, das denselben Zweck erfüllt, dann sollte man fairerweise auch die Leistung des Autors honorieren. Als Weg hierzu bietet sich, da die meisten Autoren in den Staaten leben, die Postüberweisung (*postal order*) an.

Programmvielfalt wächst ständig

Das stetig wachsende Interesse an PD-Software ließ immer neue Diskettenreihen entstehen, die ausschließlich frei kopierbare Software enthalten. Zu den bekanntesten dieser Serien zählen: **Fish**, **Faug**, **TBAG**, **RPD**, **Panorama**, **Amok**, **Franz**, **Chiron Conceptions**, **RMS**, **Taifun**, **Austria**, **Kickstart**, **ACS**, **Amicus** und **Auge**.

Nicht umsonst stehen die Fishdisks an erster Stelle dieser Auflistung. FRED FISH war es, der vor etwa dreieinhalb Jahren seine erste PD-Diskette herausgab und damit allen Interessierten eine zuverlässige Software-Quelle zugängig machte. Inzwischen hat es Fred Fish auf die stolze Zahl von 282 Disketten mit mehr als 1600 Programmen gebracht. All diese Programme sind am Ende dieses Buches in einer Liste verzeichnet. Ausführlichere Listen waren in diesem Buch nicht unterzubringen. Wer sich für weitere Listen interessiert, sei auf die Bände unserer PD-Bibliothek verwiesen, die neben

 © 1990 technicSupport

den Fishdisks auch die Reihen `Faug`, `Panorama`, `RPD`, `Franz`, `ACS`, `Austria` und `TBAG` erfassen und Informationen über Spezialserien und Diskettenmagazine geben. Hinweise dazu am Ende dieses Buches.

Apropos Spezialreihen! Unter den oben genannten Diskettenreihen dürfte die Serie `Franz` für deutsche Anwender besonders interessant sein, da alle Programme mit deutscher Anleitung versehen sind. Dies trifft auch auf die `Amok`-Reihe zu, die sich in erster Linie an Modula-2-Programmierer richtet, aber neben den Programmierbeispielen viele Programme von eigenständigem Gebrauchswert enthält. Auch auf den `Taifun`-Disketten finden sich zahlreiche Programme mit deutscher Anleitung.

Das große Programmangebot der Public Domain Serien erfüllt viele Software-Wünsche. Zwar bietet die Public Domain keine professionellen Komplettlösungen für Anwendungen wie Textverarbeitung, Desktop Publishing, CAD oder Animation, viele Programme erlauben jedoch eine Einarbeitung in diese Gebiete, und mit etwas Einfallsreichtum und Geschick lassen sich oft Ergebnisse erzielen, die sich nicht von professionell gewonnenen Resultaten unterscheiden. Zu vielen kommerziell vertriebenen Programmen gibt es in der PD Demoversionen, die zwar in einigen Fähigkeiten eingeschränkt sind, aber mit der grundsätzlichen Arbeitsweise dieser Programme vertraut machen. So gibt es z. B. eine Demoversion des Schriftsatzprogramms AmigaTEX, des Programms also, mit dem dieses Buch gesetzt wurde. Eine um deutsche Beispiele und Erklärungen erweiterte und aktualisierte Version dieser Demo ist beim Verlag erhältlich. Eine andere Art von Demos zeigt nur die Ergebnisse, die sich mit bestimmten Programmen erreichen lassen. Hierzu gehören kleine, einfallsreiche Animationen und Trickfilme, die z.B. mit `Sculpt 3D` oder `The Director` erstellt wurden und eine Bereicherung für jede Softwaresammlung sind. Neben Programmierwerkzeugen und Anwendungsprogrammen, zu denen auch verschiedenste Arten von Spielen zu rechnen sind, gibt es eine Fülle von Programmen, die die Stärken von Amiga — Grafik und Musik — unterstützen. Den eigentlichen Schwerpunkt der Amiga-Public-Domain bilden jedoch die Arbeitshilfen. Diese Utilities erleichtern die Arbeit mit Amiga in allen An-

wendungsbereichen. Innerhalb der kommerziell vertriebenen Software sucht man diese Art von Hilfsprogrammen oft vergeblich.

Bei dieser Vielfalt und Menge an Public Domain scheint es nicht verkehrt, von einem regelrechten Software-Dschungel zu sprechen. Dem Anwender einen Weg durch diesen Dschungel zu bahnen, das ist es, was wir mit unseren Büchern erreichen möchten. Weiterhelfen können aber auch die Amiga-Zeitschriften, die regelmäßig über die Public Domain berichten, oder Katalog-Disketten der PD-Anbieter.

Eine neue PD-Diskette: Was ist zu tun?

Die meisten der in Umlauf befindlichen PD-Disketten sind nicht bootfähig, der Rechner kann also nicht mit ihnen gestartet werden. Zum Starten des Rechners verwendet man daher die gewohnte Workbench-Diskette, wartet, bis die Workbench-Oberfläche geladen ist, und legt dann die PD-Disk in ein freies Laufwerk. Ist nur ein Laufwerk vorhanden, so muß man halt die Workbench-Diskette wieder herausnehmen.

Fast alle PD-Disketten sind inzwischen so aufgebaut, daß man sie von der Workbench aus erkunden kann. Das bedeutet, daß man zumindest erkennen kann, welche Programme sich auf ihr befinden, und die Anleitungen nach einem Doppelklick auf entsprechende Icons auf dem Bildschirm durchblättern kann.

Nicht jedes Programm läßt sich aber auch von der Workbench aus starten, manche müssen vor ihrer Benutzung auf bestimmte Weise installiert werden. In diesen Fällen müssen die notwendigen Arbeitsschritte aus dem CLI oder einer Shell heraus vorgenommen werden. CLI und Shell bieten dem Anwender eine weitaus leistungsfähigere Arbeitsumgebung, die jedoch nicht so einfach zu bedienen ist wie die Workbench. An dieser Thematik Interessierte seien auf mein Buch „AmigaDOS für Anwender“ verwiesen, das einen kompletten Lernkurs über das AmigaDOS enthält. An dieser Stelle können dagegen nur die allerwichtigsten Arbeitsschritte skizziert werden.

Ein häufiger Arbeitsschritt besteht darin, benötigte Programme in bestimmte Verzeichnisse der Workbench-Diskette zu kopieren. Dazu

 © 1990 technicSupport

ist mit der Workbench-Diskette zu booten. Anschließend muß ein CLI- oder ein Shell-Fenster geöffnet werden. Für das Kopieren gibt es den `Copy`-Befehl, der mit zwei Argumenten versehen werden muß: dem Namen der zu kopierenden Datei und dem Namen des Zielverzeichnisses. Dabei müssen die Namen eine gültige „Wegbeschreibung" enthalten. Die universellste, wenn auch selten kürzeste Methode besteht darin, die Wegbeschreibung (den sogenannten Pfad) mit der Laufwerksbezeichnung oder dem Namen einer eingelegten Diskette beginnen zu lassen. Um etwa das Programm `Mackie` von der in Laufwerk `df1:` liegenden Fishdisk 267 in das `c`-Verzeichnis einer Workbench-Diskette in `df0:` zu kopieren, lautet der Befehl:

```
copy df1:Mackie/Mackie df0:c
```

Wer nur ein Laufwerk besitzt, sollte mit den Namen der Disketten arbeiten, die unter den Diskettensymbolen abzulesen sind oder im CLI mit dem Befehl `info` abgefragt werden können. Für den Fall, daß die Workbench „A500 WB 1.2 D" heißt — bei Leerzeichen innerhalb der Namen müssen diese auch noch in Anführungszeichen eingeschlossen werden — nimmt der Befehl folgende Form an:

```
copy AmigaLibDisk267:Mackie/Mackie "A500 WB 1.2 D:c"
```

Auch beim Aufruf von Programmen, spielt die Wegbeschreibung eine große Rolle. Da wir nicht wissen, mit welcher Diskette ein Anwender arbeitet oder in welchem Laufwerk die Diskette liegt, findet sich in der Rubrik „Aufruf" gewöhnlich nur der Programmname. Die Wegbeschreibung muß der Anwender selbst hinzufügen. Sie ist nur dann entbehrlich, wenn das Programm bereits in das `C:`-Verzeichnis oder ein anderes mit dem `path`-Befehl (s. AmigaDOS-Handbuch) in den Suchpfad aufgenommenes Verzeichnis kopiert worden ist. Für das Programm `Mackie` kämen folgende Varianten in Frage:

```
df1:Mackie/Mackie [<Argumente>]
AmigaLibDisk267:Mackie/Mackie [<Argumente>]
```

Eine andere Variante besteht darin, zunächst mit dem `cd`-Befehl, in das Verzeichnis zu wechseln, in dem sich das Programm befindet:

```
cd df1:Mackie
```

beziehungsweise

```
cd AmigaLibDisk267:Mackie
```

Nun genügt die Eingabe des bloßen Namens

```
Mackie [<Argumente>]
```

um das Programm zu starten.

Häufig tritt das Problem auf, daß Programme noch weitere Daten benötigen und diese in einem Verzeichnis mit einem festgelegten logischen Namen suchen. Ist dieses logische Verzeichnis nicht definiert worden, erscheint entweder eine programminterne Fehlermeldung oder ein charakteristischer Fehlerrequester der Art „Please insert Volume.... in any drive". Statt der Pünktchen wird in der harten Realität die geforderte logische Bezeichnung für das Verzeichnis aufgeführt. Im Falle des Malprogramms **Vdraw** lautet der Name „Vdraw". Das Verzeichnis, das unter diesem logischen Namen angesprochen werden soll, ist meistens das Verzeichnis, in dem auch das Programm selbst liegt, auf der Panorama 19a heißt es „vdraw1.21". Die Definition des logischen Verzeichnisnamens erfolgt über den **assign**-Befehl, dem der logische Name (mit einem Doppelpunkt) sowie Weg und Name des tatsächlichen Directories folgen müssen:

```
assign vdraw: df1:vdraw1.21
assign vdraw: "panorama 19A 08Jul87:vdraw1.21"
```

Bei vielen Programmen ist es erforderlich, dem Programm mitzuteilen, was genau es zu tun hat. Dazu ist beim Aufruf der Programmname um ein oder mehrere Argumente zu ergänzen. Die Bedeutung dieser Argumente ist in den Programmbeschreibungen unter der Rubrik „Optionen", manchmal aber auch innerhalb der „Beschreibung" erklärt. Das Stichwort „Aufruf" führt alle möglichen Argumente in einer standardisierten Darstellung auf. Diese Darstellungsform soll hier noch einmal erläutert werden:

[Option]

Ein optionaler Teil ist ein Teil, der hier stehen *kann* aber nicht *muß*.

 © 1990 technicSupport

Weiterhin finden die spitzen Klammern Verwendung, welche Worte enthalten, die nicht wörtlich einzugeben sind, sondern eine Klasse möglicher Eingaben beschreiben. Dies sieht dann beispielsweise so aus:

⟨*Dateiname*⟩

Das heißt also: Hier ist das Wort **Dateiname** nicht wörtlich einzugeben, vielmehr ist *irgendein* Dateiname einzugeben.

Die folgende Darstellung besagt, daß das Wort **xyz** hier beliebig oft verwendet werden darf. Das heißt auch „Null-mal", also gar nicht.

{**xyz**}

Wenn Sie verschiedene Eingabemöglichkeiten haben, die sich gegenseitig ausschließen, so werden diese durch einen senkrechten Strich getrennt. Beispielsweise heißt das Folgende: „Hier darf entweder **a** oder **b** eingegeben werden."

a|**b**

Nun können diese verschiedenen Symbole auch kombiniert werden. So bedeutet:

muster {**x**|⟨*Dateiname*⟩}

daß nach dem Aufruf **muster** beliebig oft entweder **x** oder ein Dateiname folgen darf. Und

muster [⟨*Dateiname*⟩] [**x**|**y**]

heißt, daß hier nach dem Aufruf **muster** ein Dateiname folgen *kann*. Anschließen kann sich noch „x" oder „y", jedoch ist es hier nicht erlaubt, beide Buchstaben anzugeben, da sich die Auswahl nicht, wie im vorherigen Beispiel, wiederholen darf.

Relativ selten kommt auch noch eine Zeichenbereichsangabe vor. Die Spezifikation

a–z

besagt, daß hier *irgendein* Buchstabe zwischen **a** und **z** angegeben werden kann.

Wenn es nicht funktioniert: Typische Fehler

Welchen Aufruf Programme erfordern, läßt sich aus unseren Programmbeschreibungen entnehmen. Üblicherweise genügt hierzu die Eingabe des Programmnamens, eventuell gefolgt von einigen Optionen. Da wir nicht ausschließen können, daß sich PD-Disketten in unterschiedlichen Versionen auf dem Markt befinden, halten wir es nicht für sinnvoll, den gesamten Pfadnamen mitanzugeben. Sollte es Ihnen daher passieren, daß der Aufruf

```
<Programmname>
```

mit der Fehlermeldung

```
Unknown command <Programmname>
```

quittiert wird, so befindet sich das Programm in einem Verzeichnis, welches weder das aktuelle ist, noch zu Ihrem Suchpfad gehört. Wechseln Sie dann bitte mit dem `cd`-Befehl in dieses Verzeichnis, fügen Sie es mit

```
path add <Verzeichnisname>
```

in Ihren Suchpfad ein, oder stellen Sie dem Befehl die komplette Pfadangabe voran.

Zum Schluß noch einige Hinweise: Wer sich überhaupt nicht mit dem CLI anfreunden kann, kann zum Kopieren einzelner Dateien, aber auch für andere Arbeitsschritte das in diesem Buch beschriebene Programm `DiskMan` benutzen. Und schließlich gilt auch für die preiswerte PD-Software: Die Originaldisketten gehören an einen sicheren Ort, gearbeitet werden sollte grundsätzlich mit Kopien.

Lesehinweise

Alle Programmbeschreibungen in diesem Buch sind nach einem einheitlichen Schema aufgebaut. Das erleichtert den Umgang mit den Anleitungen und die Arbeit mit den Programmen. Es folgt nun ein Muster, das zeigt, wie die einzelnen Programmbeschreibungen in diesem Buch gegliedert sind.

© 1990 technicSupport

AUFRUF

An dieser Stelle der Programmbeschreibung wird erläutert, wie das Programm zu starten ist. Dabei ist oft auch eine Zeile für den Aufruf im CLI angegeben.

OPTIONEN

Oftmals läßt sich der Ablauf oder das Verhalten von Programmen durch die Angabe und/oder das Fortlassen von sogenannten *Optionen* (Argumenten, s.o.) steuern. Sie sind in diesem Teil der Programmbeschreibung dargestellt.

ZWECK

Bei vielen Programmen ist ihr Zweck bei erster Betrachtung nicht offensichtlich. Dies ist selbst bei solchen Programmen so, die sich schließlich als sehr nützlich erweisen. Im Teil „Zweck" werden Nutzen und Anwendbarkeit eines Programmes erläutert.

BESCHREIBUNG

In diesem meist umfangreichsten Teil wird die Bedienung eines Programmes beschrieben. Wegen seiner Länge ist dieser Abschnitt oftmals durch Zwischenüberschriften noch weiter unterteilt.

PROBLEME

Bei vielen Programmen gibt es einzelne kleine Mängel oder Gefahrenstellen, die es bei der Benutzung erfolgreich zu umschiffen gilt. Unter „Probleme" werden Sie vor solchen Klippen gewarnt.

HERKUNFT

Hier wird die Herkunft des Programmes genannt. Sie erfahren, auf welcher Diskette Sie es finden und wer der Autor ist. Auch besondere Wünsche der Autoren werden hier erwähnt.

© 1990 technicSupport

VERWEISE

Ein Programm steht selten allein. Meist gibt es andere Programme, die es ergänzen oder eine ähnliche Funktion erfüllen. Hier wird auf solche Programme hingewiesen.

Die in diesem Buch beschriebenen Programme entstammen den verschiedensten Anwendungsgebieten. Zu welcher Kategorie ein Programm gehört, läßt sich den kleinen Piktogrammen in den Kopfzeilen entnehmen. Sie haben folgende Bedeutung:

Grafikprogramme
Das Palettensymbol steht für Programme, die sich mit Bildern, Grafik und Animation beschäftigen.

Musikprogramme
Programme, die sich mit den Klangfähigkeiten des Amiga auseinandersetzen, finden sich unter dieser Rubrik.

Büroanwendungen
Der Dateikasten deutet auf typische Büroanwendungen wie Dateiverwaltung oder Tabellenkalkulation.

Spiele
Das kleine Schachbrett weist auf Spiele und andere Programme, die den Unterhaltungsaspekt in den Mittelpunkt stellen

Texte
Unter diesem Piktogramm findet man Programme zum Anzeigen oder Bearbeiten von Texten.

Druckerhilfen
Dieses Bild steht für Programme, die den Umgang mit Druckern erleichtern.

Gagprogramme
Der „schmelzende“ Bildschirm symbolisiert kleine sehenswerte Programmierspäße, die aber keinen praktischen Nutzen haben.

© 1990 technicSupport

Sonstige Arbeitshilfen
Dies ist die umfangreichste Rubrik. Sie umfaßt alle Programme, die den Arbeitskomfort steigern oder auftretende Probleme meistern.

An dieser Stelle folgen noch einige kleinere Lesehinweise, die helfen sollen, Mißverständnisse zu vermeiden. Der Begriff CLI wird benutzt, um die Arbeitsumgebung für die Verwendung von AmigaDOS-Befehlen zu bezeichnen. In diesem Fall kann nicht nur in einem CLI-Fenster, sondern auch in einem Shell-Fenster gearbeitet werden.

Das diesem Buch zugrundeliegende Taschenbuchformat macht es leider unmöglich, sehr lange Eingabezeilen darzustellen. In einigen, wenigen Fällen mußten daher Eingaben, die in der Realität in einer einzigen Zeile erfolgen m ü s s e n, auf zwei Druckzeilen aufgeteilt werden. Wir haben dies dadurch kenntlich gemacht, daß wir die zweite Druckzeile sehr weit nach rechts gerückt haben. Hier ein Beispiel aus der Beschreibung des `Dme`:

```
menuadd Spezial (LF->CRLF) (top repeat -1 (last tlate 13 down))
menuadd Spezial (CRLF->LF) (top repeat -1 (last left if c=13
                                                    del down))
```

Der Ausdruck `del down))` steht sehr weit rechts und ergibt für sich allein keinen Sinn. Wenn Sie diese Zeilen in einen Editor eingeben, müssen Sie also darauf achten, daß die letzten beiden Zeilen als eine Zeile eingeben. Das Erscheinungsbild des Textes am rechten Rand des Editors nimmt dann folgende Gestalt an:

```
...zial (LF->CRLF) (top repeat -1 (last tlate 13 down))
...zial (CRLF->LF) (top repeat -1 (last left if c=13 del down))
```

Nun aber genug der langen Vorreden. Mit diesen Informationen ausgestattet, können Sie sich jetzt auf die Anwendungsbeispiele und Programmbeschreibungen stürzen. Wir wünschen Ihnen dabei viel Spaß und viel Erfolg.

Anwendungsbeispiele

In diesem Kapitel führen wir an typischen Beispielen vor, wie Public-Domain-Programme effektiv und nutzbringend eingesetzt werden können. Die Betonung liegt hier nicht auf der ausführlichen Beschreibung einzelner Programme, sondern auf ihrer geschickten, anwendungsorientierten Kombination.

WB 1.3 für Individualisten

Bereits im Zweiten Amiga Public Domain Buch wurde an dieser Stelle die Einrichtung einer von PD-Programmen unterstützten, individuellen Workbench beschrieben. In letzter Zeit erreichten den Verlag zahlreiche Briefe von Anwendern, denen eine Übertragung der entsprechenden Arbeitsschritte auf die neue Workbench-Version 1.3 Schwierigkeiten bereitete. Aus diesem Grunde soll die Thematik der individuellen Workbench-Diskette unter dem Stichwort 1.3 noch einmal aufgegriffen werden.

Ein Problem ergab sich aus den im damaligen Beitrag verwendeten Pfadangaben bei den `Runback`-Befehlszeilen, die den Pfad relativ zum aktuellen Verzeichnis festlegten. Dieses Verzeichnis war bei der Workbench 1.2 noch das Hauptverzeichnis. Ziemlich am Anfang der `Startup-Sequence` der Workbench 1.3 steht jedoch der Befehl `cd c:`, der das `C`-Verzeichnis zum aktuellen erklärt, weshalb die Pfadangaben der später in der `Startup-Sequence` auftretenden `Runback`-Befehle dann auf ein nicht vorhandenes Verzeichnis zeigen. Dieses Problem läßt sich dadurch beseitigen, daß man den Bestandteil `c/` der Pfadangabe in `c:` ändert, wodurch ein logisches Gerät zum festen Ausgangspunkt des Pfades wird. Der Pfad hängt dann nicht mehr vom aktuellen Verzeichnis ab.

Auf welch großes Interesse die im Zweiten PD Buch vorgestellten ARP-Befehle gestoßen waren, wurde uns in der Redaktion erst bewußt, nachdem sich die Workbench 1.3 stärker verbreitet hatte. Viele Leser erkundigten sich danach, inwieweit es möglich und auch sinnvoll sei, die Befehle der Workbench 1.3 durch die ARP-Kommandos zu ersetzen. Von solchen Unternehmungen mußten wir bis vor kurzem abraten, weil sich der Großteil der Commodore-Befehle auf Grund zusätzlicher Status-Bits und weiterer Optionen so stark geändert hatte, daß die meisten ARP-Kommandos nicht mehr kompatibel waren.

Inzwischen stellt sich diese Frage jedoch nicht mehr. Ebenso wie die Workbench 1.2 gehört auch die in unserem zweiten Band beschriebene ARP-Version 1.1 der Vergangenheit an. „ARP 1.3“ heißt das neue Zauberwort! Keine Frage, daß die neuen ARP-Befehle ihren Platz auf unserer individuellen Workbench finden werden.

Die auf der individuellen Workbench 1.2 eingesetzten Programme `RunBack` (Aufrufen von Hintergrundprozessen), `Mackie` (Starten eines CLI's auf Tastendruck), `FuncKey` (Belegung der Funktionstasten mit Floskeln) und `VirusX` (Schutz gegen Virusprogramme) haben auch unter der Workbench 1.3 nichts von ihrer Bedeutung eingebüßt. Die Aufgaben von `Mackie` und `FuncKey` sollen diesmal jedoch von dem Programm `MachII` übernommen werden.

Im Falle von `ConMan` stellt sich die Frage, ob der Einsatz dieses Programmes noch sinnvoll ist, da die Workbench 1.3 ja mit `Newcon:` ebenfalls über einen Konsolen-Handler verfügt, der Tastatureingaben speichern kann. Wer `ConMan` ausschließlich wegen dieser sogenannten *History*-Funktion benutzt hat, kann sich getrost mit `Newcon:` begnügen. Wenn man darüber hinausgehende Fähigkeiten `Conmans` nutzen möchte, etwa die bequeme Handhabung der Fenster über die Funktionstasten oder das Abspeichern der gepufferten Eingaben über das `History`-Utility, sollte man `ConMan` auch auf der Workbench 1.3 installieren. Dasselbe gilt auch für die Verwendung von Conman zusammen mit der `Ash`-Shell des ARP-Projektes. In letzterem Fall ist der Einsatz von `Conman` sogar ausdrücklich zu empfehlen, weil sich mit Hilfe der `L:Conhandler`-Datei und einem entprechenden Mountlist-Eintrag das `Pip:`-Gerät anmelden läßt. Diese Pipe ist die bisher einzige, die sich als interne Pipe von der `Ash` aus nutzen läßt.

Zusätzlich zu diesen bereits bekannten Programmen werden wir die Befehle `Diskcopy` und `Format` durch das Programm `DFC2` ersetzen, sowie ein kleines Utility integrieren, das jede Tastenbetätigung mit einem Geräusch bestätigt. Hier eine Übersicht über alle Programme:

`ARP 1.3`	TBAG 31
`RunBack`	Fishdisk 65
`ConMan`	Fishdisk 165

© 1990 technicSupport

`VirusX`	Fishdisk 216
`MachII`	Fishdisk 254
`DFC2`	TBAG 29
`KeyClick`	TBAG 26

Die Strategie

Wenn man sich daran macht, die Original-Befehle der Workbench-Diskette durch andere zu ersetzen, muß man einkalkulieren, daß vielleicht das eine oder andere Programm oder eine Stapeldatei in der neuen Umgebung nicht funktioniert. Die ARP-Programmierer haben versucht, ihre Befehle so kompatibel wie irgendmöglich zu gestalten, an einigen wenigen Stellen waren Abweichungen jedoch nicht zu vermeiden. So arbeitet die **Ash** beispielsweise mit einem eigenen Resident-Befehl, der sich **Ares** nennt. Man könnte natürlich auf die Installation der ARP-Shell verzichten, um hinsichtlich des Resident-Kommandos kompatibel zu bleiben, jedoch müßte man dann eine Reihe von Vorteilen entbehren.

In diesem Kapitel werden wir jedoch den radikalen Weg gehen und so viele Original-Dateien wie möglich ersetzen, um eine maximale Leistungssteigerung zu erreichen. Sollte es dann doch einmal Probleme geben, kann man immer noch auf die Original-Workbench zurückgreifen. Das geht wahrscheinlich schneller, als wenn man lange nach der Ursache des Fehlers sucht.

Die in diesem Kapitel verwendeten Programme kann man — auch in unterschiedlichen Versionen — auf verschiedenen Disketten finden, unter anderem auch auf der Diskettenreihe zu diesem Buch. Um die Beispiele dieses Kapitels von Anzahl und Bezeichnung der Laufwerke unabhängig zu halten, finden in den Eingabezeilen die Namen der oben aufgeführten Disketten Verwendung. Der Anwender muß dann den Angaben der erscheinenden Systemrequester folgen und die jeweils vorhandenen Laufwerke mit den angeforderten Disketten füttern. Bei Verwendung anderer Disketten müssen die in den Befehlszeilen verwendeten Namen entsprechend geändert werden. Die Namen kann man unterhalb der Diskettensymbole auf der Workbench ablesen. Enthält ein Name Leerzeichen, dann muß der ge-

samte Teil der Eingabezeile, der den Namen enthält, in Anführungszeichen eingeschlossen werden. Es kann auch vorkommen, daß eine solche Diskette eine ganz andere Verzeichnisstruktur besitzt. In diesem Fall müssen die kompletten Pfadangaben geändert werden.

Auf die Verzeichnisse der zu bearbeitenden Workbench-Diskette wird über die Bezeichnungen der logischen Geräte, also `Sys:`, `C:`, `Libs:` usw., Bezug genommen. Dieses Verfahren macht es unbedingt erforderlich, daß der Amiga mit genau der Workbench-Diskette gestartet worden ist, die bearbeitet werden soll.

Aufräumen auf der Workbench

Wie nicht anders zu erwarten, muß auch die Workbench 1.3 von überflüssigem Ballast befreit werden, bevor sich weitere Programme auf ihr unterbringen lassen. Zu diesem Zweck fertigt man eine Kopie der Workbench-Diskette an, startet den Rechner mit dieser Diskette neu und begibt sich in das CLI, wozu ein Doppelklick auf das `Shell`-Symbol genügt.

Die folgenden Arbeitsschritte löschen Programme, die so spezielle Funktionen haben, daß sie in den meisten Fällen entbehrlich sind. Leser, die auf eines dieser Programme keinesfalls verzichten wollen, müssen den entsprechenden Arbeitsschritt übergehen.

Reichlich Platz läßt sich dadurch schaffen, daß man die komplette `Utilities`-Schublade löscht:

```
delete sys:Utilities all
delete sys:Utilities.info
```

Nur selten vermissen wird man die Programme `FixFonts`, `InitPrinter`, `NoFastMem` und `MergeMem` aus dem `System`-Verzeichnis:

```
cd sys:system
delete (fi|no|me|in)#?
```

Für die meisten Anwendungen sind auch die Zeichensätze überflüssig, die mit

```
delete fonts:#? all
```


© 1990 technicSupport

gelöscht werden können. Wer nicht auf die Zeichensätze verzichten will, kann jedoch die eigentlichen Zeichensatzdateien auf eine andere Diskette auslagern. Dieses Verfahren ist in der Beschreibung des Programmes `Ren` ausführlich erörtert.

Die Installation der ARP-Befehle

Die ARP-Befehle beanspruchen wesentlich weniger Platz als die Commodore-Befehle. Bei der Installation werden die alten Befehle überschrieben, d. h. durch die neuen ersetzt. Man sollte sich also nicht darüber wundern, daß nach dieser Installation mehr Platz auf der Workbench-Diskette vorhanden ist als vorher.

Die ARP-Befehle der Version 1.3 werden mit einem sehr komfortablen Installationsprogramm geliefert. Dieses Programm namens `ArpInstall` läßt sich am einfachsten mit einem Doppelklick von der Workbench aus starten. Auf verschiedenen Bildschirmseiten wird der Anwender nach seinen Wünschen bezüglich der Installation gefragt. Gehen wir die einzelnen Abschnitte hintereinander durch:

- Die zwei Seiten mit einleitenden Informationen werden mit jeweils einem Klick auf [Continue] übersprungen.
- Die Frage nach der Installation der `Arp.library` wird mit [Yes] beantwortet. Die Voreinstellung des Zielverzeichnisses auf `Libs:` ist korrekt und wird mit der [Return] -Taste bestätigt. Mit [Proceed] wird die Installation gestartet.
- Auch die Frage nach der Installation der Kommandos wird mit [Yes] beantwortet. Die folgende Seite zeigt alle Standard-Befehle an, die bereits für die Installation angewählt sind. Verfolgt man keine individuellen Absichten, kann man diese Seite mit [OK] verlassen, das auf `C:` voreingestellte Zielverzeichnis mit [Return] bestätigen und die Installation mit [Proceed] starten. An einem Balken läßt sich ablesen, wie weit die Installation bereits fortgeschritten ist.
- Auf dieselbe Art und Weise verfährt man mit den hierauf folgenden erweiterten Kommandos.

- Die letzte Frage nach der Installation der ARP-Shell (**Ash**) wird ebenfalls bejaht, das Zielverzeichnis `L:` mit [Return] bestätigt und die Installation mit [Proceed] in Gang gesetzt.
- Mit [Exit] wird das Installationsprogramm verlassen.

Möchte man die Installation bereits jetzt überprüfen, so muß man mit

```
ed s:startup-sequence
```

den Editor aufrufen um die `Startup-Sequence` zu ändern. Dabei interessiert zunächst die Zeile

```
resident CLI L:Shell-Seg SYSTEM pure add; Shell aktivieren
```

die in

```
resident CLI L:Ash SYSTEM pure add
```

geändert werden muß. Anschließend ist der Editor mit der Tastenfolge [Esc] [X] [Return] zu verlassen, wobei die `Startup-Sequence` in ihrer neuen Form gespeichert wird.

Nun muß mit dieser Diskette ein Neustart durchgeführt werden. Ein Anklicken des `Shell`-Symboles öffnet ein Fenster mit dem bekannten Titel „AmigaShell“, durch den man sich nicht täuschen lassen sollte. Daß es sich in Wirklichkeit um die ARP-Shell handelt, kann durch Eingabe eines Fragezeichens mit anschließendem [Return] ermittelt werden. Die Arp-Shell listet dann eine Reihe bereits implementierter Befehle — ein Verhalten, das die Commodore-Shell nicht zeigt. Fenster, die nicht über das `Shell`-Piktogramm, sondern über den `NewShell`-Befehl geöffnet werden, tragen übrigens den neuen Titel „AShell“.

Die Installation der übrigen Programme

Am Kopieren der schon aus Band II bekannten Programme auf die Workbench-Diskette hat sich kaum etwas geändert, abgesehen von den Diskettennamen:

```
copy AmigaLibDisk65:RunBack/RunBack c:
```


© 1990 technicSupport

```
copy AmigaLibDisk216:VirusX/VirusX  c:
copy AmigaLibDisk165:ConMan/Conman c:
copy AmigaLibDisk165:ConMan/Conhandler L:
copy AmigaLibDisk165:ConMan/Conlib.lib libs:Conhandler.library
```

An dieser Stelle ist auf eine Abweichung zwischen der hier benutzten `ConMan`-Version 1.3 und der Version 1.1 von Fishdisk 133 hinzuweisen. Die `Conhandler.library` heißt auf der Fishdisk 165 `conlib.lib` und muß daher in den richtigen Namen umbenannt werden. Aus diesem Grund unterscheidet sich die obenstehende Zeile von der in Band II angeführten: Sie sorgt neben dem Kopieren zugleich für die Änderung des Dateinamens. Die Verwendung der alten `Runback`-Version hat seinen Grund darin, daß wir dieses Kommando später zur Beschleunigung der `Startup-Sequence` resident machen wollen, die neueren Versionen von Fishdisk 152 bzw. Fishdisk 214 für das residente Laden mit `Ares` aber nicht geeignet sind.

Kommen wir zu den neuen Programmen: Das Kopierprogramm `DFC2` wird am besten zusammen mit seinem Icon in der System-Schublade untergebracht, von wo es sich später auch mit der Maus aufrufen läßt:

```
copy "Disk of the Month #29:Utilities/DFC2#?" sys:System
```

Auch das Programm `KeyClick` kann in diesem Verzeichnis untergebracht werden.

```
copy "Disk of the Month #26:Utilities/Key#?" sys:System
```

Für `MachII` bietet sich hingegen wieder das `C`-Verzeichnis an:

```
copy AmigaLibDisk254:MachII/MachII.UnReg C:MachII
copy AmigaLibDisk254:MachII/SetMach C:
```

An dieser Stelle empfiehlt es sich, eine Sicherkeitskopie von der neuen Workbench anzufertigen. Falls bei den folgenden Arbeitsschritten ein Fehler unterläuft, muß das zeitraubende Kopieren dann nicht wiederholt werden.

Der Startvorgang

Die Vorteile der neuen Diskette möchte man natürlich gleich nach dem Bootvorgang nutzen. Die Aktivierung der verschiedenen Programme und Geräte muß also bereits beim Startvorgang erfolgen. Bei der Original-Workbench sorgt dafür die Datei `S:Startup-Sequence`, die eine weitere Datei namens `S:StartupII` aufruft. Diese zweite Datei arbeitet nicht mehr mit einem CLI sondern mit der AmigaShell und kann daher die Zeitvorteile der residenten Befehle nutzen.

Dieselbe Aufteilung in zwei Dateien ist auch bei unserer individuellen Workbench erforderlich, jedoch wird sich der Aufbau beider Dateien deutlich von den ursprünglichen Stapeldateien unterscheiden, so daß es verwirrend wäre, die folgende Beschreibung auf Änderungen zu stützen. Einfacher ist es, die Dateien neu anzulegen.

```
delete s:startup-sequence
ed s:Startup-sequence
```

Mit dem Editor erstellt man die neue `Startup-Sequence`, die lediglich aus fünf Zeilen besteht:

```
system/fastmemfirst
Addbuffers df0: 10
c:SetPatch >NIL:
AShell from s:Quickstartup
endcli >nil:
```

Das `FastMemFirst`-Kommando sorgt dafür, daß der wertvolle CHIP-Speicher geschont wird. Wer nur 512 KByte besitzt, kann diesen Befehl weglassen. Dem Laufwerk `Df0:` wird ein kleiner Puffer spendiert, mit `Setpatch` werden einige Fehler im Kickstart-ROM korrigiert. Und nun das Entscheidende: `AShell` aktiviert die ARP-Shell und öffnet ein Shell-Fenster, in dem eine weitere Stapeldatei `s:Quickstartup` abgearbeitet werden soll. Ein Befehl wie

```
resident CLI L:Ash SYSTEM pure add
```

ist nicht mehr erforderlich. Für die Aufnahme des Shell-Befehlsprozessors in den System-Teil der Resident-Liste sorgt der `AShell`-Befehl automatisch.

 © 1990 technicSupport

Mit [Esc] [X] [Return] wird der Editor unter Abspeichern der `Start-up-Sequence` verlassen, um dann erneut für das Editieren der Datei `S:Quickstartup` aufgerufen zu werden:

```
ed s:Quickstartup
```

Beim Abtippen der folgenden Zeilen müssen die Zeilennummern unbedingt weggelassen werden!

```
 1 BindDrivers
 2 SetClock load
 3 FF >NIL: -O
 4 system/setmap d
 5 ares c:ares c:makedir c:assign c:Newshell c:Prompt force
 6 ares c:cd c:dir c:list c:copy c:rename c:delete c:runback force
 7 makedir ram:t ram:env ram:clipboards
 8 assign T: ram:t ENV: ram:env CLIPS: ram:clipboards
 9 conman >nil: -c
10 mount pip:
11 runback >nil: -4 c:MachII
12 runback >nil: -4 c:Virusx
13 runback >nil: -3 sys:system/keyclick
14 path ram: c: sys:system s: sys:prefs add
15 LoadWB delay
16 endcli >NIL:
```

Verlassen wird der Editor wieder über [Esc] [X] [Return].

Besitzt man keine Erweiterungen, die mit `Binddrivers` eingebunden werden müssen, dann kann Zeile 1 entfallen. Ist keine akkugepufferte Uhr vorhanden, kann Zeile 2 durch

```
Date ?
```

ersetzt werden. Zeile 3 sorgt für eine schnellere Textausgabe auf dem Bildschirm, Zeile 4 aktiviert die deutsche Tastaturbelegung.

In den Zeilen 5 und 6 wird eine Reihe von Befehlen resident geladen, was bedeutet, daß diese Befehle im Speicher bereit stehen. Bei der Arbeit mit diesen Befehlen ist die Workbench-Diskette nicht mehr erforderlich, zudem wird die Ausführung dieser Befehle extrem beschleunigt. Sind zum Beispiel die Befehle `AShell` und `Prompt`

resident geladen, läuft das Öffnen eines Shell-Fensters wesentlich schneller ab. Durch Erweiterung dieser Zeilen können auf Wunsch auch noch weitere Befehle in residentem Zustand bereitgestellt werden. Dabei sollte man beachten, daß die ARP-Shell bereits über einige residente Befehle verfügt. Diese ein weiteres Mal zu laden, ist überflüssig und kostet wertvollen Speicherplatz.

In dieser und auch in den beiden folgenden Zeilen, wird ein Vorteil der ARP-Befehle deutlich: Oftmals können sie mehrfache Argumente verarbeiten, auch wenn die vergleichbaren Commodore-Befehle dazu nicht in der Lage sind. Zeile 7, die einige Verzeichnisse in der Ram-Disk einrichtet, müßte mit dem Commodore-`Makedir` in drei Zeilen aufgeteilt werden:

```
makedir ram:t
makedir ram:env
makedir ram:clipboards
```

ARP ist da schneller und übersichtlicher. Zeile 8 weist die soeben angelegten Verzeichnisse logischen Geräten zu, die vom Betriebssystem verlangt werden. In Zeile 9 wird das Standard-Ein-/Ausgabegerät `Con:` durch `Conman` ersetzt.

Zeile 10 dient dem Anmelden des `Pip:`-Gerätes. Auf Wunsch können in dieser Zeile weitere Geräte wie `Aux:`, `Speak:`, die Commodore-Pipe `Pipe:` oder auch `Newcon:` angemeldet werden:

```
mount pip: aux: speak: pipe: newcon:
```

In den Zeilen 11–13 werden unsere Hilfsprogramme aktiviert, d. h. sie werden mit Hilfe des Programmes `RunBack` als Hintergrundprozeß gestartet. Zeile 14 legt den Suchpfad fest, der alle Verzeichnisse enthalten sollte, in denen sich ausführbare Programme befinden. Die letzten beiden Zeilen laden die Workbench und schließen das Shell-Fenster.

Wenn man das `Pip:`-Gerät nicht benötigt, kann man Zeile 10 natürlich auch ganz fortlassen. Beläßt man diese Zeile jedoch im oben angegebenen Zustand, dann wird es erforderlich, die `Mountlist` um einen entsprechenden Eintrag zu erweitern. Dazu lädt man die

 © 1990 technicSupport

`Mountlist` mit dem Befehl `ed devs:mountlist` in den Editor, bewegt sich mit der Schreibmarke an das untere Ende dieser Datei und ergänzt sie um die folgenden Zeilen:

```
/*
     Conman Pipe für die ARP-Shell
*/
PIP:        Handler = L:Conhandler
            Stacksize = 600
            Priority = 5
            GlobVec = 0
#
```

Anschließend wird der Editor wie gewohnt verlassen.

Ein erster Testlauf

Für einen ersten Probelauf muß Amiga zurückgesetzt und mit der neuen Diskette gestartet werden. Bei diesem ersten Start meldet sich `MachII` mit einem Requester „Can't open Configuration File" (kann Konfigurationsdatei nicht öffnen), der sich durch Anklicken des [Cancel] -Gadgets entfernen läßt.

Bevor die Workbench-Oberfläche erscheint, sollten jetzt drei kleine Fenster der Programme `VirusX`, `MachII` und `KeyClick` auftauchen. In der System-Schublade müßten die Icons der Programme `KeyClick` und `DFC2` zu sehen sein. Ob die residenten Befehle funktionieren, läßt sich durch die Eingabe von

```
dir ram:
```

feststellen. Bei diesem Befehl darf kein Zugriff auf das Laufwerk erfolgen, das Lämpchen muß also dunkel bleiben. Hat man das `Pip:`-Gerät angemeldet, kann man mit dem `assign`-Befehl überprüfen, ob das Gerät auch in dem `Devices:`-Teil der Liste angezeigt wird.

Um die von `MachII` geforderte Konfigurationsdatei anzulegen, ist das schmale `MachII`-Fenster, in dem abwechselnd Speicherplatz und Uhrzeit gezeigt werden, mit einem Klick der linken Maustaste zu aktivieren, bevor es ein weiteres Mal, diesmal aber mit der rechten

Maustaste angeklickt wird. Es öffnet sich das Kontrollfeld, in dem sich alle Funktionen des Programmes einstellen lassen. Fürs erste soll es genügen, wenn wir auf Tastendruck ein Shell-Fenster erhalten. Wir klicken mit der linken Maustaste in das Eingabefeld der fünften Zeile, löschen dessen Inhalt und tragen dafür den Ausdruck

```
AShell con:0/10/640/246/AShell
```

ein. `MachII` ist so voreingestellt, daß automatisch das Fenster aktiviert wird, auf dem sich der Mauszeiger befindet. Dieses doch sehr gewöhnungsbedürftige Verhalten kann durch Anklicken des [SunMouse] -Feldes abgestellt werden. Weitere Einstellmöglichkeiten lassen sich aus der Beschreibung des Programmes entnehmen. Verlassen wird das Kontrollfeld über den [Save] -Button. `MachII` legt unter dem Namen `S:Mach.config` seine Konfigurationsdatei ab, in der auch die neu eingetragene Funktionen gespeichert sind.

Durch Niederhalten der linken Amiga-Taste und gleichzeitiges einmaliges Drücken von [Esc] sollte sich jetzt ein Fenster mit dem Titel „AShell" öffnen lassen. Wenn dies einmal nicht geschieht, kann es daran liegen, daß beim Booten der PAL-Bildschirm nicht erkannt wurde. Der Bildschirm hat dann nur 200 Zeilen, was man daran erkennt, daß sich der Mauszeiger nicht vollständig nach unten bewegen läßt. Unser Shell-Fenster ist für diesen Bildschirm zu groß; der Bootvorgang ist in diesem Fall zu wiederholen.

Wurde auf das Anmelden des `Newcon`-Gerätes verzichtet, so ruft ein Anklicken des Shell-Piktogramms auf der Workbench einen Fehlerrequester hervor — mit der sinnreichen Meldung „Please insert volume NEWCON in any drive (Bitte legen Sie die Diskette NEWCON in ein Laufwerk)". Das `Shell`-Piktogramm ruft das `CLI`-Programm in der System-Schublade auf, das wiederum das Fenster über den Konsolen-Handler `Newcon:` öffnen möchte. Man braucht hier nur auf [Cancel] zu klicken, dann wird anstelle von `Newcon:` auch `Con:` — in unserem Fall gleichbedeutend mit `Conman` — akzeptiert. Am `CLI`-Programm liegt es auch, daß das Fenster den Titel „AmigaShell" zugewiesen bekommt, obwohl sich hinter ihm die ARP-Shell verbirgt.

© 1990 technicSupport

Dieses kleine Problem läßt sich aber rasch aus der Welt schaffen, indem man dafür sorgt, daß über das Shell-Piktogramm nicht das `CLI`-Kommando der `System`-Schublade, sondern der `AShell`-Befehl aufgerufen wird. Dazu wechselt man auf die Workbench, klickt das Shell-Symbol einmal mit der linken Maustaste an und ruft anschließend die Funktion `Info` aus dem `Workbench`-Menü auf. In dem erscheinenden Fenster klickt man das `Default Tool`-Eingabefeld an, ändert den Eintrag in `C:AShell` ab und verläßt das Fenster über den [Save] -Button. Beim nächsten Doppelklick auf das Shell-Symbol funktioniert dann alles nach Wunsch.

Exkurs: AmigaShell und ConMan:

Anwender, die wegen eventueller Inkompatibilitätsprobleme doch lieber mit der AmigaShell weiterarbeiten, aber dennoch nicht auf `Conman:` verzichten wollen, können diese Arbeitsumgebung auf Tastendruck erhalten, wenn sie in das entsprechende Eingabefeld des `MachII`-Kontrollfensters den statt des `AShell`-Kommandos den Befehl:

```
NewShell con:0/10/640/246/AmigaShell
```

eintragen. Etwas mehr Mühe bereitet es, den störenden „Plaease insert Volume Newcon...“-Requester beim Anklicken des Shell-Piktogramms zu unterdrücken. Hier wird es notwendig, den `CLI`-Befehl aus der `System`-Schublade zu ändern. Dazu sind keine tiefschürfenden Programmierkenntnisse nötig. Die Veränderungen können durch einfaches Überschreiben eines Textes im fertigen Programm vorgenommen werden. Eine solche nachträgliche Manipulation eines bereits lauffähigen Programms bezeichnet man als *Patchen*.

Mit Absicht habe ich hier den mit einem negativen Beiklang belegten Begriff „Manipulation“ verwendet. Patchen kann in vielen Fällen hilfreich sein, es wird aber auch oft mißbraucht. Da werden Texte oder Parameter in fremden Programmen überschrieben, die dann als eigene geistige Leistung ausgegeben werden, obwohl das Patchen allgemein nicht sehr viel Genialität erfordert.

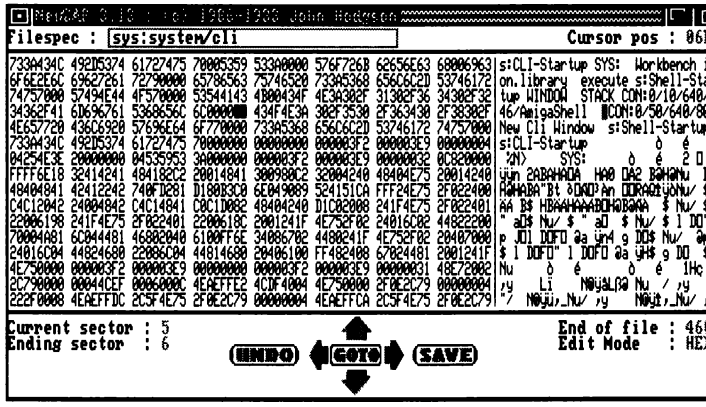
Beim Patchen eines Programmes sollte man sich über zweierlei bewußt sein: Fehlerhaftes Patchen kann die Funktionstüchtigkeit eines Programmes bis zur völligen Unbrauchbarkeit in Mitleidenschaft ziehen, daher sollten solche Versuche nur an Kopien vorgenommen werden. Auch besitzt man keinerlei Rechte an einem gepatchten Programm und sollte solche auch dann nicht weiter verbreiten, wenn sie zur frei kopierbaren Software gehören. Dies führt nämlich dazu, daß irgendwann vergessen wird, daß ein Programm gepatcht wurde. Fehler oder andere Abweichungen werden dann dem Urheber des Programmes angelastet, der sich selbst wiederum nicht erklären kann, weshalb sich sein Programm plötzlich anders verhält als gewohnt. Wenn also Programme gepatcht werden, dann darf es in der Regel nur für den eigenen Gebrauch geschehen.

Zum Patchen eignet sich grundsätzlich jeder Disketten-Monitor, bequemer geht es aber mit Monitoren, die sich nicht an den physikalischen Strukturen einer Diskette (Spuren und Blöcke), sondern an Dateien orientieren. Ein solches Programm ist der an anderer Stelle in diesem Buch vorgestellte **NewZap** von der Fishdisk 164. Lesen Sie bitte dessen Beschreibung, bevor Sie die folgenden Anweisungen in die Tat umsetzen. Nach dem Aufruf dieses Programms lädt man die Datei `Sys:System/Cli` und „blättert" in Vorwärtsrichtung, bis man im Sektor 5 angekommen ist. Im rechten der beiden großen Fenster erblickt man die Textfolge „NEWCON:....", die den Konsolenhandler für die Shell festlegt. Mit dem Mauszeiger klickt man direkt auf das erste „N", **NewZap** sollte dann als Cursorposition den Wert 52 anzeigen. Jetzt kann man an dieser Stelle den gewünschten Text über die Tastatur eingeben. Dabei ist zu beachten, daß **NewZap** die deutsche Tastaturbelegung unberücksichtigt läßt. Um einen Doppelpunkt zu erhalten muß man ein „Ö" eingeben, für einen Schrägstrich ein „-". Man trägt ein:

```
CON:0/10/640/246/AShell
```

Da dieser Text dort beginnen muß, wo vorher das Wort „NEWCON" begann, unser Text aber kürzer ist, endet diese Passage auf „..AmigaShellll". Die beiden überflüssigen „l" könnten durch zwei Leerzeichen ersetzt werden, die dann aber auch im Titel der Shell-

 © 1990 technicSupport



Das CLI-Kommando wird gepatcht

Fenster angezeigt würden. Eleganter ist es da, den Hex-Wert 0 einzugeben, was nur innerhalb des linken Fensters erfolgen kann. Zwei Zeichen im linken Fenster entsprechen immer einem Zeichen im rechten. Man klickt also im linken Fenster auf die Zeichenkombination, die dem ersten überflüssigen „l“ entspricht (im rechten Fenster kann man sehen, ob die Position richtig ist), und gibt viermal die [0] ein. Anschließend speichert man die geänderte Datei mit [Save] ab. Hat man alles richtig gemacht, wird beim nächsten Anklicken des Shell-Icons ein Shell-Fenster mit `Conman`-Unterstützung aufgerufen, das sich über den ganzen Bildschirm erstreckt, da die Fenster-Parameter ebenfalls geändert wurden.

Noch mehr Platz

Wenn alles funktioniert, ist es an der Zeit für eine weitere Sicherheitskopie. Anschließend lassen sich noch einige nicht mehr benötigte Dateien entfernen. Zunächst einmal sind das die Befehle, die bereits in der ARP-Shell enthalten sind — mit zwei Ausnahmen: Die

Befehle `run` und `execute` werden von vielen Programmen genutzt und oft auch nur dann gefunden, wenn sie im `C:`-Verzeichnis stehen. Deshalb sollte man sie besser nicht löschen.

```
cd c:
delete endif endskip lab else skip if quit failat
```

Im `L:`-Verzeichnis lassen sich das `Shell-Seg` sowie — falls nicht benötigt — der `Newcon-Handler` löschen:

```
cd L:
delete Shell-Seg Newcon-Handler
```

Aus der `System`-Schublade können `Format` und `Diskcopy` samt Icons entfernt werden. Ihre Aufgabe wird von `DFC2` übernommen:

```
cd sys:system
delete format#? diskcopy#?
```

Auch im `S:`-Verzeichnis kann Platz gespart werden:

```
cd s:
delete startupii startup-sequence.hd
delete spat dpat cli-startup
```

Die Datei `Shell-Startup` wird beim Öffnen jedes neuen Shell-Fensters abgearbeitet und enthält einige mit `alias` definierte Befehle. Einige sind nur Krücken, die mehr schlecht als recht versuchen, den Commodore-Befehlen Fähigkeiten zu verleihen, die die ARP-Befehle von Hause aus besitzen. Über die Brauchbarkeit der anderen läßt sich streiten. Möchte man sie löschen, lädt man die Datei mit

```
ed s:shell-startup
```

in den Editor und löscht die entsprechenden Zeilen. Die erste Zeile, die die Gestalt des Eingabepromptes festlegt, sollte man stehen lassen. Arbeitet man häufiger mit Programmen, die einen größeren Stack als die voreingestellten 4000 Byte benötigen, ist es praktisch, die gewünschte Stackgröße in dieser Datei zu definieren, da sie dann von jedem Shell-Fenster automatisch übernommen wird. Aus Geschwindigkeitsgründen sollte der Befehl `Stack` in diesem Falle schon in der `Startup-Sequence` resident geladen werden.

 © 1990 technicSupport

Wie weit die neue Workbench jetzt gefüllt ist, hängt davon ab, wie genau man den einzelnen Schritten dieser Programmbeschreibung gefolgt ist. Der genaue Wert läßt sich mit dem Befehl

```
info
```

ermitteln, er dürfte bei knapp 70% liegen. Trotz der vielen Verbesserungen stehen also noch etwa 30% der Diskette für die Aufnahme weiterer Programme zur Verfügung. Vielleicht warten Ihre Lieblingsbefehle und -programme ja schon in der Diskettenbox auf ihren Einsatz? Gewiß finden sich auch in diesem Buch viele nützliche Utilities, aus denen Sie Ihre **ganz** individuelle Workbench 1.3 zimmern können.

© 1990 technicSupport

Die PD-Story

Ein Paar tiefschwarz umrandeter, noch halb geschlossener Augen näherte sich, vom Kaffeeduft geleitet, dem Frühstückstisch. Für Erhard Köber begann der Tag wieder einmal mehrere Stunden zu früh. „Frische Bohnen?“, fragte er mit geschlossenen Augen, während seine rechte Hand instinktsicher die Thermoskanne fand. „Nein, aber frischer Kaffee!“, antwortete Christiane, was ihr einen fassungslosen Blick ihres Mannes einbrachte: „Kannst Du solche Spitzfindigkeiten nicht wenigstens am frühen Morgen unterlassen?“ — „Klingt fast so, als ob du wieder einmal schlecht geschafen hast“, äußerte sich Christiane. „Schlecht ist etwas untertrieben“, murmelte Erhard, „ich hab schon wieder diesen Alptraum gehabt — all diese Vögel, die in mein Arbeitszimmer fliegen und kleine Häufchen auf meinem Amiga hinterlassen. Außerdem bin ich erst etwa um 4.00 Uhr ins Bett gekommen!“ Mißmutig kaute er auf seinem Marmeladenbrötchen herum.

Eine Weile herrschte Schweigen, denn Christiane traute sich nicht, ihn nach seinen nächtlichen Erfolgen zu befragen. „Hätte ich mir bloß nie diesen [redacted] Computer gekauft!“, platzte es schließlich aus Erhard Köber heraus. „Drei Stunden lang habe ich versucht, das Druckformular mit einem Seitenvorschub zu versehen, aber mit diesem ulkigen Ed scheint das unmöglich zu sein... aber davon verstehst du ja sowieso nichts. Übrigens, wenn Du heute einkaufen gehst, könntest Du ja mal bei Data-Frust vorbeigehen, vielleicht haben die einen PD-Editor, der das kann — mit dem Med habe ich das nämlich auch nicht geschafft.“ Verzweifelt trank er seine Tasse leer, um dann in aller Eile in das Labor zu fahren, in dem er nun schon seit einigen Jahren als Zahntechniker arbeitete.

Christiane frühstückte in aller Ruhe zu Ende, brachte etwas Ordnung in die Küche und versorgte das acht Monate alte Töchterchen Sabine mit frischen Windeln. Schließlich setzte sie sich vor Amiga, um die Fortschritte ihres Mannes zu begutachten. Sofern es der Nachwuchs erlaubte, nutzte sie ihren Mutterschaftsurlaub, um sich

in die Materie, die ihr einst so fremd gewesen war, einzuarbeiten. Im Gegensatz zu Erhard bereitete ihr diese Beschäftigung immer mehr Freude. „Er nimmt das alles viel zu ernst“, interpretierte sie die Entwicklung der letzten Wochen, die ihr aus zahlreichen Frühstücksgesprächen hinreichend bekannt war.

Begonnen hatte alles im Dezember 1987, als Erhard einen Teil seines Weihnachtsgeldes für einen Amiga 500 hinblätterte. Dieses Gerät hatte ihn fasziniert, seit er es bei seinem Freund Giovanni in voller Aktion erlebt hatte. Mit Hilfe Goivannis machte er schnelle Fortschritte, und bald schon wuchs in ihm der Wunsch, seinen Computer in irgendeiner Weise nutzbringend einzusetzen. Gelegenheit dazu bot sich im Juni 1988 auf der Hauptversammlung seines Sportvereins, wo er leichtsinnigerweise versprach, einen Teil der Buchhaltung zu übernehemen — mit EDV-Hilfe versteht sich. Leichtsinnigerweise deshalb, weil Giovanni kurz vorher nach Italien zurückgekehrt und Erhard auf sich allein gestellt war. Nun suchte er verzweifelt nach einer preiswerten Lösung für eine Dateiverwaltung. Bei einem der seltenen, weil teuren Telefonate mit Giovanni hatte ihn dieser auf das Zweite Public Domain Buch des Verlages technicSupport aufmerksam gemacht. Dort sollte die Lösung eines ähnlichen Problems beschrieben sein. Lachend machte Giovanni noch einige dumme Bemerkungen wie „Lies das Buch und erkenne dich selbst!“, mit denen Erhard überhaupt nichts anfangen konnte.

Mit dem Kauf des Buches nahm das Unheil seinen Lauf. Zunächst erzürnte Erhard Köber sich über die Namen der in der PD-Story beteiligten Personen, die dem seiner Frau und seinem eigenen zur Verwechslung ähnlich waren. Die Herausgeber hätten doch wenigstens eine Bemerkung wie „frei erfunden“ oder „Ähnlichkeiten mit lebenden Personen rein zufällig“ hinzufügen können. Als ihn auch noch sein Systemfachhändler mit der spitzen Bemerkung „Guten Tag, Herr Körber, wie geht es denn ihrer Briefmarkensammlung?“ begrüßte, feilte er zwei Abende lang an einem Beschwerdebrief an den Verlag, bis ihn Christiane von der Albernheit dieser Unternehmung überzeugte.

Ebenso rasch war er aber auch von der Public Domain überzeugt und hatte es sich in den Kopf gesetzt, seine Datenbank mit eben die-

 © 1990 technicSupport

ser frei kopierbaren Software zu erstellen. Auf den Lösungsweg aus der PD-Story des Zweiten Amiga Public Domain Buches konnte und wollte er jedoch aus mehreren Gründen nicht zurückgreifen. Zum einen wollte er sich nicht noch mehr mit Gerhard Körber über einen Kamm scheren lassen, zum anderen erschien ihm das Verfahren zu undurchsichtig und unattraktiv, um bei seinen Vereinskameraden Eindruck zu machen. Außerdem erhoffte er sich im Stillen, daß seine Frau die Eingabe der Daten übernehmen würde. Da sie aber nichts von Computern verstand, mußte er dazu eine besonders einfache Lösung finden.

„Wie hat er doch damals gejubelt, als er dieses Programm in den Listen gefunden und kurz angetestet hatte", erinnerte sich Christiane, während sie die Hyperbase-Diskette zur Hand nahm. „Eine Komplettlösung..., Workbench..., Intuition.., Editor ade!" war er in der ersten Euphorie auf einem Bein um den Eßtisch herumgetobt. Die Ernüchterung folgte rasch, als er feststellen mußte, daß er für die Erstellung von Masken und Formularen doch wieder einen Editor brauchte. Das Problem, daß er keine deutschen Sonderzeichen eingeben konnte, schob er erst einmal auf die lange Bank, zur Not würden es auch die Umschreibungen tun.

Christiane betrachtete inzwischen in einem Editor die Datei „Mahnung.prt". Es handelte sich um einen Mahnbrief, der als Serienbrief an alle säumigen Vereinsmitglieder gehen sollte. Auf dem Bildschirm sah das folgendermaßen aus:

```
1),10,5,1,_____,{2},,
2),11,5,1,_____________________________________,{4}&" "&{3},,
3),12,5,1,____________________________,{5},,
4),13,5,1,_________________________________,{6}&" "&{7},,
5)Berlin\, d. 12.2.1989,16,45,1,,,,
6)Liebes Clubmitglied\,,19,1,1,,,,
7)mit diesem Schreiben möchten wir Sie daran erinnern\, daß
                                                  Sie,21,1,1,,,,
8)sich mit Ihren Beitragszahlungen im Rückstand befinden.,
                                                   22,1,1,,,,
9)Ihre Beitragszahlungen sind erfüllt bis zum Monat,24,1,
  1,____________,{18}[3]&{18}[4]&".19"&{18}[1]&{18}[2]&".",,
```

```
10)Wir möchten Sie daher bitten\, den ausstehenden Betrag
                                                  von,25,1,1,,,,
11)DM ,27,1,1,___,(12+12*(88-({18}[1]&{18}[2]))-({18}[3]&
                                              {18}[4]))*{17},,
12)auf unser Vereinskonto zu überweisen.,29,1,1,,,,
13)Mit freundlichen Grüßen,31,1,1,,,,
14)Der Vorstand,34,1,1,,,,
```

(*Anm. der Red.: Die Zeilennummern wurden der besseren Übersicht halber hinzugefügt. Sie gehören nicht in das ursprüngliche Formular.*)

Auch Christiane konnte mit diesen Zahlen zunächst nichts anfangen. Sie verließ den Editor wieder und druckte sich zunächst die Maske „Verein.fmt“ aus, um die Bedeutung der einzelnen Datenfelder überblicken zu können. Die Maske hatte folgende Gestalt:

```
1)Vereinsverwaltung,1,26,1,,,,
2)Anrede: ,3,3,1,,,__________^,
3)Name: ,5,3,1,,,______________________________^,
4)Vorname: ,7,3,1,,,___________________^,
5)Straß e: ,9,3,1,,,__________________________^,
6)PLZ: ,11,3,1,,,####^,
7)Ort: ,11,14,1,,,___________________________^,
8)Tel: ,11,55,1,,,________________^,
9)**************************************************
  *************************,13,3,1,,,,
10)Bank: ,15,3,1,,,______________________________^,
11)BLZ: , 15,41,1,,,########^,
12)Kto-Nr:, 15,56,1,,,_______________^,
13)**************************************************
   **************************,17,3,1,,,,
14)Abteilung: ,19,3,1,,,_______________^,
15)Status: ,19,32,1,,,______^,
16)Geb.: ,19,50,1,,,##.##.####^,
17)Monatsbeitrag: DM,21,3,1,,,###^,
18)Bezahlt bis: ,21,25,1,,,##.##^,
```

Neugierig geworden, rief sie Hyperbase auf, lud die Datei „Verein.dat“ sowie das Formular „Mahnung.prt“. Sie ließ einen Daten-

© 1990 technicSupport

satz suchen und druckte ihn aus. Von dem Ergebnis war sie positiv überrascht:

```
      Frau
      Claudia Keller
      Weberweg 22
      1000 Berlin 456


                                    Berlin, d. 12.2.1989


Liebes Clubmitglied,


mit diesem Schreiben möchten wir Sie daran erinnern, daß Sie
sich mit Ihren Beitragszahlungen im Rückstand befinden.

Ihre Beitragszahlungen sind erfüllt bis zum Monat 01.1987.
Wir möchten Sie daher bitten, den ausstehenden Betrag von

DM 420

auf unser Vereinskonto zu überweisen.

Mit freundlichen Grüßen


Der Vorstand
```

„Da kann er doch eigentlich ganz zufrieden sein“, dachte sie und machte sich an die Analyse des Formulars. Die ersten Zeilen waren einfach zu durchschauen, sie sorgten dafür, daß die jeweilige Anschrift ab der zehnten Zeile zu Papier gebracht wird. Offensichtlich wollte Erhard Fensterbriefe verwenden, weil er immer noch keinen

© 1990 technicSupport

Traktor für seinen Drucker besaß, mit dem er Etiketten ausdrucken könnte. Stutzig wurde Sie in Zeile 9, wo es hieß:

```
9)Ihre Beitragszahlungen sind erfüllt bis zum Monat,24,1,
  1,____________,{18}[3]&{18}[4]&".19"&{18}[1]&{18}[2]&".",,
```

Aus der Maske konnte Sie ersehen, daß in Datenfeld 18 der Stand der Beitragszahlungen erfaßt ist. Zunächst wurde ihr nicht recht einsichtig, warum dort in den ersten zwei Stellen das Jahr und dann erst der Monat stehen sollte. Nach einer Weile kam sie schließlich darauf, daß Erhard ja sicherlich mit der Search-Funktion alle säumigen Zahler herausfiltern wollte. Da diese Funktion keine Verknüpfung zweier Datenfelder zuließ, mußte er Monat und Jahr wohl oder übel in einem einzigen Feld unterbringen. So konnte er jetzt in Feld 18 als Suchoperator „<8810“ eingeben und alle Datensätze finden, deren Beitragszahlungen sich nicht bis zum Oktober 1988 erstreckten. Hätte er für Feld 18 aber das Format MM-JJ gewählt, hätte er mit „<1088“ suchen müssen, wobei ihm alle Datensätze, die bis zum November oder Dezember irgendeines Jahres bezahlt hätten, entgangen wären. Für den Computer ist „1286“ nun einmal größer als „1088“. Auch Zeile 11 wurde jetzt verständlich: Die ersten beiden Zahlen stellten das aktuelle Datum (Monat und Jahr) dar. Der so kompliziert erscheinende Ausdruck bewirkte nichts weiter, als die Differenz zum Stand der Beitragszahlungen in Monaten zu berechnen und das Ergebnis mit der Höhe eines Monatsbeitrages (Feld 17) zu multiplizieren.

„Ordentliche Arbeit!“, lobte sie im Stillen ihren Mann, um sich gleich wieder zu wundern: „Warum wertet er diese Datenfelder so umständlich aus? Statt `18[3]&18[4]` könnte er doch viel einfacher `18[3,4]` schreiben.“ Um es auszuprobieren, holte sie ihre Disketten, die sie in der hintersten Ecke ihrer Schublade versteckt hielt, kopierte Erhards Diskette und änderte das Formular nach ihren Vorstellungen ab, um festzustellen, daß Hyperbase mit dieser Anweisung nichts anfangen konnte, obwohl sie in der „.doc“-Datei aufgeführt war. Ganz offensichtlich stieß sich Hyperbase an dem Komma. Sie hoffte in den Masken und Formularen der beiden Beispieldateien „mail“ und „records“, die sich auf der Fishdisk befanden, solch eine Anweisung zu finden, um die Syntax überprüfen zu

 © 1990 technicSupport

können — eine vergebliche Hoffnung. Umso mehr freute sie sich für Erhard, daß er einen Ausweg aus diesem Dilemma gefunden hatte. Jetzt fehlte wirklich nur noch das Steuerzeichen mit dem ASCII-Wert 12, damit jeder Brief auch auf einer neuen Seite beginnt...

„Was willst Du denn noch mit der CedDemo", fragte Freddy, während er die RPD 040 kopierte, „Du hast doch neulich schon die Professional-Version gekauft?". — „Die soll Erhard erst zu Weihnachten bekommen", antwortete Christiane, „aber er braucht nun einmal schon jetzt einen Editor, mit dem er Steuerzeichen eingeben kann. Er ist gerade dabei, ein Formular für Hyperbase zu erstellen." Es entwickelte sich eine Fachsimpelei, gegen die Sabine nach einiger Zeit mit lautem Gebrüll intervenierte. Christiane gab sich geschlagen, bezahlte die Diskette und machte sich mit ihrer Tochter auf den Heimweg.

Zu Hause angekommen, zog sie es vor, zunächst die leidige Hausarbeit zu erledigen. Bis vor kurzem hatte sie Erhard dabei tatkräftig unterstützt, aber zur Zeit arbeitete er bis zu 15 Stunden pro Tag, da mußte sie halt alleine damit fertig werden. Glücklicherweise waren sie beide keine pedantischen Ordnungsfanatiker. Bevor sie das Abendessen zubereitete, begutachtete sie schnell noch die CedDemo. „Die Beschränkung auf 5 KByte lange Texte würde Erhard nicht stören", stellte sie fest. „Deutsche Umlaute kann man auch nicht eingeben, aber mit den Sonderzeichen — das funktionierte einwandfrei..."

Um 19 Uhr klingelte das Telefon. „Tut mir leid, Schatz, es wird heut wieder später", vernahm sie die leise Stimme ihres Mannes. Vorsichtshalber ging sie auf Distanz zum Telefonhörer: Erfahrungsgemäß würde jetzt die obligatorische Schimpfarie auf Norbi und seine großartige Gesundheitsreform folgen...

Kurz vor 21 Uhr kam Erhard nach Hause. „Fünfmal mußte ich um den Block fahren, bis ich einen Parkplatz gefunden hatte", fluchte er vor sich hin, während er seinen Mantel in die Ecke schmiß. Erst beim Abendessen besserte sich seine Laune. „Hast Du einen geeigneten Editor bekommen?", fragte er mit vollem Mund. „Freddy hat mir eine Diskette kopiert, Du kannst ja mal gucken, was drauf ist!"

Christiane stellte sich dumm. „War denn viel los bei Data-Frust?“, äußerte sich Erhard, weil ihm nichts anderes einfiel. „Es ging, es war nur ein Kunde da, der sich beschwerte, weil ein Spiel nicht auf seinem Rechner laufen wollte.“ — „Amiga?“, hakte Erhard nach. — „Jaja, Freddy hat sowas erzählt, daß Commodore die 500er jetzt mit neuen Kickstart-RAMs oder so ähnlich ausliefere.“ — „ROMs, mit o wie Otto“, verbesserte Erhard, „wird ja wieder ein nettes Chaos geben! Na ja, mal sehen, was Du mir für einen Editor angeschleppt hast!“ Er erhob sich und ging in sein Arbeitszimmer.

Eine gute Stunde später ertönte lautes Jubelgeheul: „Super, dieser Editor! Und ganz leicht zu bedienen!“ Er zerrte Christiane vor den Monitor. „Hier, dieses L in reverser Darstellung — das Ei des Kolumbus! Man kann zwar nicht in derselben Zeile weiterschreiben, wenn man es eingegeben hat, aber es funktioniert trotzdem!“ Er startete Hyperbase und steuerte ohne nachzudenken die Print-Funktion an. Nach dem dritten ausgedruckten Blatt wurde ihm klar, daß er in seinem Übereifer die Anweisung gegeben hatte, sämtliche Datensätze, zur Zeit 84 Stück, auszudrucken. Kurzerhand schaltete er die Anlage aus. Er wußte, daß er diese Nacht besser schlafen würde.

Das Wochenende stand ganz im Zeichen der Weihnachtsvorbereitungen. Wie jedes Jahr bestand Erhard darauf, seine „vielgerühmten“ Plätzchen und Lebkuchen zu backen, und wie jedes Jahr hinterließ er die Küche in einem apokalyptischen Chaos. Christiane revanchierte sich, indem sie sich an das Klavier setzte und drei Stunden lang Weihnachtslieder übte. Erst am Sonntag Abend setzte Erhard die friedliche Adventsstimmung aufs Spiel und sich an den Amiga. „Ich schreibe nur noch schnell eine Einladung zur Jahreshauptversammlung im Februar, mit dem neuen Editor ist das ja ganz einfach.“ Christiane ahnte nichts Gutes. Bald schon würde er merken, daß sein hochgelobter Editor keine deutschen Umlaute verkraftet. Zu Christianes Verwunderung blieb es im Arbeitszimmer jedoch verdächtig ruhig. „Wie läuft es denn so?“, erkundigte sie sich schließlich nach Erhards Wohlbefinden. „Fast fertig“, erwiderte ihr Mann mit abgeklärter Stimme, „der neue Editor kann natürlich keine deutschen Umlaute, aber ich benutze ihn jetzt nur für die Steuerzeichen, den eigentlichen Text erfasse ich mit dem Ed — falls

© 1990 technicSupport

Dir das etwas sagen sollte.“ Zehn Minuten später erklärte ein lautes, zufriedenes Gähnen den 4. Advent für beendet.

Am Montag fällte Christiane die Entscheidung: Weihnachten würde sie ihre Karten auf den Tisch legen und Erhard mit ihren Computerkenntnissen überraschen. Sie war sich nicht sicher, ob er so schnell verzeihen würde, daß sie ihn die ganze Zeit hinter das Licht geführt hatte. Um ihn zu besänftigen, plante sie, ihm eine spezielle Hyperbase-Diskette einzurichten und noch eine Lösung für das Problem mit den Umlauten zu finden.

Die Einrichtung der Hyperbase-Diskette begann mit dem Abspecken einer Workbench-Diskette. Christiane zog eine Kopie, begab sich in das CLI und begann, alles zu löschen, was ihr überflüssig erschien. Ein sicheres Todesurteil für die Schubladen `Trashcan`, `Expansion`, `fonts`, `Demos` und `Utilities`. Nach Eingabe des Befehles „info“, wußte sie, daß nur noch 75% der Diskette belegt waren. Sie setzte ihre Arbeit fort, indem Sie eine ganze Reihe von Befehlen aus dem `c`-Verzeichnis löschte. Übrig blieben nur noch `addbuffers`, `cd`, `copy`, `delete`, `dir`, `edit`, `endcli`, `execute`, `failat`, `info`, `loadwb`, `makedir`, `path`, `rename`, `run` und `setclock`. Ebenso löschte sie sämtliche Dateien aus dem `System`-Verzeichnis bis auf `Cli`, `Diskcopy` und `Setmap`. Diese Aktion brachte weitere 24% Platzgewinn. Schließlich entfernte sie aus dem `devs`-Verzeichnis die `mountlist` sowie das `narrator.device` und das `serial.device`, aus dem `devs/keymaps`-Directory alle Dateien bis auf die deutsche Tastaturbelegung `d`. Zum Schluß entfernte Sie noch alle überflüssigen Druckertreiber aus dem `devs/printers`-Ordner. Noch einmal tippte Sie den „info“-Befehl: Nur noch 36% der Diskette waren belegt. „Das sollte eigentlich ausreichen“, fand Christiane und tätigte den Klammeraffengriff, um ihre Kreation einmal auszuprobieren.

`Unknown command echo`. Christiane hatte vergessen einen Blick auf die Startup-Sequence zu werfen, die natürlich noch Befehle enthielt, die sie inzwischen gelöscht hatte. Sie startete das System mit einer kompletten Workbench, legte die abgemagerte Version in das Zweitlaufwerk und rief aus dem CLI den `Ed` auf:

```
ed df1:s/startup-sequence
```

Die ersten Zeilen sahen wie folgt aus:

```
echo "A500 Workbench 1.2  Version 33.56  23-APR-87*N"
BindDrivers
if EXISTS sys:system
    path sys:system add
endif
if EXISTS sys:utilities
    path sys:utilities add
endif
```

Da gab es ja noch einiges zu tun: Die ersten beiden Zeilen würde sie löschen. Auf die Ausgabe der Versionsnummer konnte sie nun wirklich verzichten, und auch der `Binddrivers`-Befehl war überflüssig, da keine Hardware-Erweiterung vorhanden war, die diesen Befehl benötigte. Die `if/endif`-Abfragen waren auch unnötig, die Einbindung der `utilities`-Schublade in den Suchpfad mußte ebenfalls entfernt werden, da diese ja gelöscht worden war. Nach einigen weiteren Änderungen war die `Startup-Sequence` wesentlich kompakter geworden:

```
path sys:system add
SetMap d
Addbuffers df0: 20
LoadWb
failat 30
SetClock >NIL: Opt load
endcli > nil:
```

Jetzt funktionierte es. Christiane legte Fishdisk 131 in das Zweitlaufwerk, um das Hyperbase-Programm zu übertragen:

```
makedir df0:hyperbase
cd df1:
copy hyperbase.info df0:
cd hyperbase
copy hyperbase df0:hyperbase
```

Jetzt suchte sie noch ein passendes Tool-Icon, um den Start von der Workbench zu ermöglichen. Auf Fishdisk 151 wurde sie fündig.

© 1990 technicSupport

`DiskSellect` sollte zwar eine Diskettenbox darstellen, würde aber auch als Karteikasten durchgehen.

```
copy df1:icons/Tool_Icons/DiskSellect.info
df0:Hyperbase/Hyperbase.info
```

Die Diskette war immer noch zu 63% frei. Sie kopierte zu Testzwekken Erhards Datei auf ihre Diskette und ordnete mit Hilfe der `Snapshot`-Funktion alle Fenster und Icons übersichtlich an. Vorsichtshalber machte sie noch eine Sicherheitskopie.

Jetzt mußte sie nur noch das Problem mit den Umlauten in Angriff nehmen. Eine Idee war ihr beim letzten Friseurbesuch gekommen. Dort bleibt ja immer genug Zeit zum Nachdenken. Man müßte die Sonderzeichen nur in der deutschen Umschreibung eingeben, also „ae“, „oe“, „ue“, „ss“ usw. und — wegen der Eindeutigkeit — ein weiteres Zeichen voranzustellen. Ein Backslash wäre ideal, da er in normalen Texten nicht vorkommt. Jetzt müßte sie nur noch einen Weg finden, um diese Zeichenkombinationen in der Datei „Verein.dat“ automatisch ersetzen zu lassen. Christiane dachte an den Zeileneditor `Edit`, der im Dos-Handbuch so miserabel dokumentiert ist, daß seine Existenz schnell wieder vergessen wird. `Edit` manipuliert bereits bestehende Dateien und legt das Ergebnis in einer zweiten Datei ab. Die nötigen Befehle müssen dazu in einer dritten Datei stehen.

Christiane begann mit dieser dritten Datei:

```
ed df1:hyperbase/Umlaute
```

Sie benötigte nur einen einzigen Befehl: `ge` (*global exchange*=überall ersetzen):

```
ge /\ae/ä/
ge /\oe/ö/
ge /\ue/ü/
ge /\Ae/Ä/
ge /\Oe/Ö/
ge /\Ue/Ü/
ge /\ss/ß/
```

Mit [Esc], [x] und [Return] verließ sie `Ed`, rief Hyperbase auf und gab zur Probe einige Datensätze ein, die Namen wie `M\ueller` oder `Strau\ss` enthielten. Sie verließ die Datenbank und vergewisserte sich mit dem `cd`-Befehl, daß sie sich auch im `hyperbase`-Verzeichnis befand. Jetzt rief sie den Zeileneditor auf:

```
edit Verein.dat to testdatei with Umlaute
```

Auf dem Bildschirm häuften sich die Fehlermeldungen über abgeschnittene Zeilen, und Christiane sah sich gezwungen, die Dateien „Verein.dat" und Testdatei näher anzusehen. Zunächst fiel ihr auf, daß die Zeilen der Datei „Verein.dat" sehr lang waren — teilweise über 200 Spalten. Die Zeilen der Testdatei waren wesentlich kürzer, die Datensätze unvollständig. Die Substitution der Umlaute war allerdings korrekt ausgeführt worden. Die Ursache hierfür fand sie im DOS-Handbuch: `Edit` ist auf maximal 120 Zeichen voreingestellt, glücklicherweise gab es aber eine Option, um diesen Wert zu ändern. „300 Zeichen sollten ausreichen", dachte sie und versuchte es noch einmal:

```
edit Verein.dat to Testdatei with Umlaute opt w300
```

Diesmal klappte alles, und Christiane gönnte sich ihre wohlverdiente Kaffeepause.

Ein wenig bequemer wollte Sie es Erhard schon machen. Dazu erstellte sie unter dem Namen „Ersetzen" eine kurze Batchdatei, die sie ebenfalls im Hyperbase-Verzeichnis abspeicherte:

```
echo "Lege Sicherheitskopie verein.bak an"
copy verein.dat verein.bak
echo "Ändere Umlaute: Bitte warten"
edit verein.bak verein.dat with withfile opt w300
echo "Vorgang beendet!"
```

Christiane ärgerte sich, daß sie den `Echo`-Befehl gelöscht hatte und ihn nun doch in das c-Verzeichnis zurückkopieren mußte. Ihr Ärger verflog rasch, als sie nach Eingabe des Befehls

```
execute Ersetzen
```


© 1990 technicSupport

feststellte, daß alles zu ihrer Zufriedenheit verlief. Wie geplant enthielt „Verein.dat“ nun die richtigen deutschen Umlaute. Sie wollte sich gerade daran machen, diese Batchdatei mit Hilfe des PD-Programms `Xicon` von der Workbench startfähig zu machen, als Sabine lauthals nach Nahrung verlangte. Christiane blickte zur Uhr. Es war ohnehin schon spät geworden, und bald würde Erhard nach Hause kommen. Rasch packte sie alles zusammen...

Ausnahmsweise erschien Erhard pünktlich zum Abendessen. „Endlich mal vor Ladenschluß Feierabend gehabt“, kommentierte er den Blumenstrauß, den er seiner Frau mitgebracht hatte. Nach dem Abendessen igelte er sich in seinem Arbeitszimmer ein, um die Weihnachtsgeschenke zu verpacken. Nachdenklich blickte er auf das Buch „Amiga für Einsteiger“, das er für seine Frau vorgesehen hatte. „Irgendwie müßte man doch auch sie für diese Sache interessieren können“, dachte er, „und wenn nicht, dann wird sie sich wenigstens über die Theaterkarten und den schon bestellten Babysitter freuen.“ Es würde eine recht interessante Bescherung geben...

© 1990 technicSupport

Programmbeschreibungen

Dieses Kapitel bildet den Hauptteil des vorliegenden Buches. In alphabetischer Ordnung werden 51 Public Domain Programme aus den unterschiedlichsten Anwendungsgebieten ausführlich und praxisorientiert beschrieben.

AmiCron

Startet Programme zu bestimmten Zeiten

AUFRUF

Das Programm wird in einem CLI-Fenster durch den Befehl

`AmiCron24a` ⟨*Dateiname*⟩

gestartet. `AmiCron` benötigt ein eigenes CLI-Fenster, in das die zu startenden Programme ihre Ausgaben leiten können.

ZWECK

`AmiCron` dient dazu, Programme zu festgelegten Zeitpunkten zu starten.

BESCHREIBUNG

Bevor man mit dem Programm arbeiten kann, muß es installiert werden. Zunächst einmal ist das Programm selbst in das `c`-Directory der Workbench zu kopieren, damit es später problemlos aufgerufen werden kann. Anschließend muß eine Datei erzeugt werden, in der die Daten über die von `AmiCron` zu startenden Programme mit den erforderlichen Zeitangaben abgelegt werden. Im Gegensatz zu der früheren Version muß diese Datei *nicht mehr* nach Unix-Manier in einem Verzeichnis namens `Sys:usr/spool/cron` (arrgh!) liegen. Fehlermeldungen werden jetzt im `t:`-Verzeichnis abgelegt, wo eine Datei namens `CronErr` erzeugt wird. Neben diesen Anpassungen an die Amiga-Umgebung wurde ein Fehler bei der Auswertung des Monats beseitigt sowie einige programmtechnische Verbesserungen vorgenommen.

Wird `AmiCron` mit dem bloßen Programmnamen aufgerufen, müssen die Daten in einer Datei namens `CronTab` abgelegt werden, die sich im `S:`-Verzeichnis befinden muß. Zu beachten ist hier der Doppelpunkt, gemeint ist also das logische Verzeichnis. Wird dem Programmaufruf jedoch ein Dateiname (ggf. mit Pfad) mitgegeben, so

werden die Daten aus dieser Datei gelesen. Der Aufruf könnte also auch

```
AmiCron Ram:Zeitplan
```

lauten.

Festlegen der Zeitpunkte

Eine solche Datei kann mit jedem Editor erstellt werden. In ihr wird festgelegt, welches Programm zu welchen Zeitpunkten gestartet werden soll. Jede Anweisung muß in einer separaten Zeile stehen. Die Anweisung muß linksbündig sein, darf also nicht eingerückt werden. Jede dieser Zeilen muß wiederum aus sechs Einzelangaben bestehen, die durch Leerzeichen voneinander zu trennen sind. Hier die Bedeutung der Felder und die Werte, die sie annehmen können:

1. Minute (0–59)
2. Stunde (0–23)
3. Tag im Monat (1–31)
4. Monat (1–12)
5. Wochentag (0–6), wobei 0 für Sonntag, 6 für Samstag steht.
6. Name des auszuführenden Programms

Steht in einem Feld ein Sternchen (`*`), wird der entsprechende Parameter nicht berücksichtigt. In jedem Feld kann auch eine Folge von Zahlen stehen, die durch Kommata oder Bindestriche voneinander getrennt sind. Die folgenden Beispiele verdeutlichen die Möglichkeiten, die sich hieraus ergeben.

„`* * * * * dir`“ gibt jede Minute das aktuelle Verzeichnis aus.

„`30 6 * * 1-5 play musik:WakeUp`“ Spielt jeden Morgen um 6.30 Uhr das Musikstück „WakeUp“, nur nicht am Samstag und Sonntag.

„`0 0 29 2 0,6 delete ram:#?`“ löscht den Speicher in jedem Schaltjahr, wenn der 29.Februar zusätzlich auf einen Samstag oder Sonntag fällt.

© 1990 technicSupport

Nach der Eingabe solcher Einträge kann die Datei unter dem Namen „crontab“ abgespeichert werden. Nun kann man mit dem Befehl `Newcli` ein neues CLI-Fenster öffnen und dort den Befehl `AmiCron` eingeben. Zu den angegebenen Zeiten wird `AmiCron` dann seine Aufgaben erfüllen. So kann man sich etwa jeden Samstag um 18 Uhr an den Beginn der Sportschau erinnern lassen oder — Modem mit Selbstwählfunktion vorausgesetzt — zu nachtschlafender Zeit eine Datenfernübertragung starten.

Möchte man `AmiCron` bei jedem Systemstart aktivieren, ist in die `Startup-Sequence` der folgende Befehl einzufügen, der ein neues CLI öffnet und in diesem Fenster eine Batchdatei ausführt.

```
run newcli con:0/0/640/50/CronFenster s:startcron
```

Größe und Titel des Fensters sowie der Name der Batchdatei können natürlich den eigenen Vorstellungen angepaßt werden. Die Batchdatei „startcron“ muß im `s:`-Verzeichnis abgelegt werden und braucht nur einen einzigen Befehl enthalten:

```
AmiCron
```

Das Ganze macht natürlich nur dann Sinn, wenn man das System jedesmal mit derselben Diskette startet oder eine Festplatte besitzt.

HERKUNFT

Das UNIX-Programm `Cron` stammt von STEVE R. SAMPSON. An den Amiga angepaßt hat es

```
Rick Schaeffer          UUCP:  seismo!uunet!iscuva!ricks
E. 13611 26th Ave.      Phone: (509)928-3533
Spokane, WA  99216, USA
```

Die Überarbeitung zur Version 2.4 ist der SOFTWARE BREWERY zu verdanken, genauer gesagt:

```
Christian Balzer
Im Wingertsberg 45
D-6108 Weiterstadt
```

Fundquelle ist die Panorama-Diskette 026d.

ARP 1.3

Doping für die Workbench

AUFRUF

Die ARP-Befehle der Version 1.3 werden mit einem komfortablen in Modula-2 geschriebenen Installationsprogramm, `ArpInstall`, geliefert, das durch einen doppelten Mausklick von der Workbench aus gestartet wird. Die Befehle selbst werden bis auf wenige Ausnahmen unter denselben Namen aufgerufen wie die ursprünglichen Commodore-Befehle.

ZWECK

Die Befehle des `ARP 1.3` ersetzen viele Befehle der Workbench 1.3 durch neu geschriebene Kommandos, die genauso zu handhaben sind wie ihre AmigaDOS-Vorväter. Sie belegen jedoch weniger Platz auf Diskette und im Speicher, sind schneller, erlauben einen umfangreicheren Einsatz von *Wildcards* (Jokern) und besitzen so manch praktische Zusatzoption.

BESCHREIBUNG

Der Begriff ARP (*AmigaDOS Resource Project*) steht für die Bestrebungen einer Gruppe amerikanischer Programmierer um CHARLIE HEATH, Komfort und Leistungsfähigkeit des AmigaDOS entscheidend zu verbessern. Die Abkehr von der usprünglichen, den Commodore-Entwicklern nicht gerade schmeichelnden Namensgebung *AmigaDOS Replacement Project* könnte auf taktische Gründe zurückzuführen sein. Vielleicht erhofft man sich eine gute Zusammenarbeit mit Commodore, die dazu führen könnte, daß ARP einmal zum Standard wird: ARP-Kommandos auf der Workbench, die `ARP.library` im Kickstart — ein schöner Traum, vielleicht zu schön, um wahr zu werden!

Hatten doch bereits die zur Workbench 1.2 kompatiblen ARP-Befehle der Version 1.1 (besprochen im Zweiten Amiga Public Domain

 © 1990 technicSupport

Buch, S. 168–176) binnen kurzer Zeit eine großer Zahl begeisterter und zufriedener Anwender hinter sich versammelt. Schon bald darauf erschien die Workbench 1.3 mit vielen Neuerungen wie FastFileSystem, Shell und neuen Schutzbits, die von den **ARP**-Befehlen zwangsläufig noch nicht berücksichtigt wurden. Man war gezwungen, das **ARP**, dessen Komfort man gerade eben schätzen gelernt hatte, wieder aufzugeben.

Auf den ersten Blick konnte man sich auch mit der neuen Workbench anfreunden: FastFileSystem, Resident- und Alias-Kommando, Nutzung des Script-Bits aus der Shell heraus usw. bedeuteten ja eine erhebliche Verbesserung gegenüber der Workbench 1.3, auch waren manche Befehle tatsächlich ein wenig kürzer geworden. Im Vergleich zum **ARP** jedoch besaßen sie immer noch eine unerträgliche Länge. Befehle wie **Type**, **Cd** oder **Rename** akzeptierten nach wie vor keine Jokerzeichen. Solch eine im Grunde elementare Fähigkeit konnte ihnen zwar — und das auch nur bedingt — unter Verwendung der Stapeldateien **Spat** und **Dpat** aufgepfropft werden, über die Ausführungsgeschwindigkeit sollte man aber besser den Mantel des Schweigens breiten. Dunkel blieb auch der Sinn der Befehle **Setenv** und **Getenv**, nicht etwa weil sie Umgebungsvariable einrichten, sondern weil die Art und Weise, mit der sie diese Aufgabe verrichten, auch mit den Befehlen **Echo** und **Type** realisiert werden kann. Ein modernes Betriebssystem verlangt geradezu nach einem Standard für Umgebungsvariable, nach dem sich dann auch die Programmierer von Anwendungsprogrammen richten können, statt auf die Methode zu verfallen, ihren Programmen vom Anwender gewünschte Parameter nur über die Tool Types-Einträge in den zugehörigen Icons zu übermitteln. Dabei hatte sich mit dem **Set**-Befehl des Manx C längst ein heimlicher Standard herausgebildet, der bereits von einigen Anwendungsprogrammen und auch vom **ARP 1.1** unterstützt wurde. Kurz und gut: Man nutzte die Wartezeit nach jedem Booten des Amiga für ein kleines Stoßgebet und erflehte eine zur Workbench 1.3 kompatible **ARP**-Version.

Die Gebete wurden erhört! **ARP 1.3** liegt endlich vor: Auf der TBAG-Diskette Nr. 31. Die neuen Kommandos verstecken sich in dem Installationsprogramm **ArpInstall**, welches bislang leider erst von

zwei kürzeren Doc-Dateien begleitet wird. Die ausführliche Dokumentation der Einzelbefehle ist noch in Arbeit. Besonders tragisch ist dies aber nicht, da man mit fast allen ARP-Befehlen so arbeiten kann, wie man es von den Commodore-Befehlen her gewohnt ist, und die wichtigsten Verbesserungen und Unterschiede schon in der überblicksartigen Dokumentation der TBAG 31 zur Sprache kommen.

Die Installation der ARP-Befehle kann in unserem Anwendungsbeispiel über die individuelle Workbench 1.3 nachgelesen werden. An dieser Stelle soll lediglich noch einmal darauf hingewiesen werden, daß die Installation in drei Varianten erfolgen kann. Begnügt man sich mit den Standard-Befehlen, dann kann man im AmigaDOS so weiterarbeiten, wie man es bislang gewohnt war, ohne sich allzuviele Gedanken darüber zu machen, mit welchen Befehlen man es eigentlich zu tun hat. Man nutzt dann im wesentlichen die Platzersparnis und die Geschwindigkeitsvorteile der ARP-Befehle. Entscheidet man sich darüber hinaus für die Installation der ARP-Shell (ASH), so sollte man auch die Startup-Sequence ändern und sich mit den besonderen Eigenschaften der ARP-Shell auseinandersetzen. Anwender ohne grundlegende Kenntnisse des AmigaDOS, die ohnehin nur selten mit dem CLI oder der AmigaShell arbeiten, sollten auf diesen Schritt verzichten. Dasselbe gilt auch für die Erweiterungskommandos, deren Installation hingegen in jedem Fall zu empfehlen ist, wenn man sich zur Nutzung der ASH entschlossen hat: Dies gilt besonders für die Befehle AShell und ARes.

Die Standard-Befehle des ARP

Die Hauptverbesserungen setzen natürlich in erster Linie im C-Verzeichnis der Workbench-Diskette an. Die vergleichende Übersicht zeigt deutlich, daß die ARP-Befehle in fast allen Fällen um etwa 20 bis 75 Prozent kürzer sind als die vergleichbaren Commodore-Kommandos. Sie verbrauchen weniger Platz auf der Diskette, sind schneller geladen und gehen auch mit dem Rechnerspeicher sehr sparsam um. In den Genuß dieses Vorteils kommt man also, ohne eine einzige neue Befehlssyntax erlernen zu müssen. Trotz ihrer Kürze sind die ARP-Befehle aber auch noch leistungsfähiger!

 © 1990 technicSupport

ARP und Workbench 1.3: Dateilängen im Vergleich

Befehl	DOS 1.3	ARP 1.3
AddBuffers	876	492
Ask	648	592
Assign	3008	1264
Avail	1964	692
Binddrivers	2920	—
Break	956	460
CD	1756	580
ChangeTaskPri	1072	488
Copy	9848	2784
Date	4208	1092
Delete	6124	936
Dir	8772	2200
DiskChange	680	392
DiskDoctor	6892	—
Echo	992	432
Ed	19564	—
Edit	18164	—
Else	860	508
EndCLI	696	256
EndIf	40	40
EndSkip	40	40
Eval	1640	920
Execute	4712	2444
Failat	1028	376
Fault	2688	1512
FF	3236	—
FileNote	692	344
GetEnv	916	336
IconX	3884	—
If	2536	968
Info	2068	1312

© 1990 technicSupport

Befehl	DOS 1.3	ARP 1.3
Install	2436	984
Join	1056	564
Lab	40	40
List	9972	2500
LoadWB	2784	—
Lock	1012	—
Makedir	768	396
Mount	5432	2372
NewCLI	2788	1068
NewShell	2752	1068
Path	2136	784
Prompt	584	396
Protect	1396	928
Quit	1036	408
Relabel	872	460
RemRAD	304	—
Rename	632	968
Resident	2692	—
Run	2568	400
Search	6916	1520
SetClock	4972	—
SetDate	2652	832
SetEnv	836	448
SetPatch	3844	—
Skip	1204	696
Sort	1868	1143
Stack	872	372
Status	1772	804
Type	2284	1412
Version	2680	712
Wait	1424	628
Which	2104	904
Why	576	420

© 1990 technicSupport

So sind zum Beispiel auch die Befehle `Type`, `Cd`, `Sort`, `Search`, `Join`, `Protect`, `Path` und `Rename` in der Lage, Jokerzeichen zu verarbeiten; `Assign` und `Mount` verkraften mehr als ein Argument: Befehlszeilen wie

```
rename #?.txt #?.tex
cd dh0:l#?
assign libs: sys:libs s: sys:s l: sys:l
mount aux: speak: pipe:
```

werden erst durch diese Eigenschaften möglich. Manche Befehle verfügen über recht praktische Zusatzoptionen: `Type` etwa versieht eine Datei mit einer Kopfzeile, wenn die Option `Opt B` verwendet wird, `Opt F` schickt am Ende einer Datei ein *Formfeed* (Seitenvorschub) an das Ausgabegerät. Das `Sort`-Kommando hängt sich bei einer großen Quelldatei nicht gleich auf, wenn der Stack zu klein bemessen ist. Die Aufzählung ließe sich fortsetzen.

Optimiert wurde auch die Ausführungsgeschwindigkeit. `Search` arbeitet zum Beispiel dreimal so schnell, `Sort` sogar fünfmal so schnell wie der entsprechende Commodore-Befehl. Darüber hinaus besitzen alle `ARP`-Kommandos eine erweiterte Online-Hilfsfunktion: Wird als Argument ein Fragezeichen übergeben, erfolgt die Ausgabe des Befehlsformates in der gewohnten Form, bei Eingabe eines weiteren Fragezeichens erhält man eine noch ausführlichere Hilfestellung.

Die Joker des ARP

Die `ARP`-Befehle bieten diverse Möglichkeiten zur Verwendung von Jokerzeichen, die von den AmigaDOS-Befehlen nicht unterstützt werden. Bei der Verwendung solcher Möglichkeiten, in Stapeldateien etwa, muß man sich jedoch darüber bewußt sein, daß diese Batchdateien mit den offiziellen Befehlen nicht mehr funktionieren:

- Das `*`-Zeichen besitzt dieselbe Funktion wie die AmigaDOS-Zeichenfolge `#?`.
- Zeichen, die in eckige Klammern eingeschlossen werden, werden als alternative Möglichkeiten betrachtet. Der Ausdruck `*.[ch]` deckt sich also mit allen Dateien, die auf `.c` oder `.h` enden. Mit

einem Bindestrich werden sogar Bereichsangaben möglich: `type [a-e]*` gibt alle Dateien aus, die mit einem der Buchstaben von `A` bis `E` beginnen.

- Die Dateien, die sich mit einem Suchmuster decken, lassen sich aber auch von der Bearbeitung ausschließen: `List ~*.info` listet alle Files, die nicht auf `.info` enden.

Gründe für die Effektivität

Derartige Leistungssteigerungen bei einer drastischen Verkürzung der Programmlänge sind natürlich nur deshalb möglich, weil die Commodore-Befehle selbst alles andere als optimal programmiert wurden. Während die Entwickler von Metacomco mit der Programmiersprache BCPL arbeiten, die einen recht aufgeblähten Code erzeugt, wurden die `ARP`-Kommandos in Assembler programmiert, gewissermaßen in der Muttersprache des Prozessors. Ungeachtet der Frage, welche Programmiersprache denn nun die beste ist — für ein Betriebssystem ist Assembler immer die erste Wahl, zumal der Gesichtspunkt der Portabilität hier keine Rolle spielt. Software-Häuser werden das etwas anders sehen, da sie natürlich daran interessiert sind, einmal entwickelte Prozeduren ohne große Anstrengungen auch für die Entwicklung anderer Betriebssysteme oder Programme nutzen zu können.

Häufig benutzte Funktionen wurden in die `Arp.library` ausgelagert, wo sie dann allen Befehlen zur Verfügung stehen, die auf diese Funktionen zugreifen müssen. Die `Arp.library` dient aber nicht nur der Platzersparnis, sondern erlaubt es auch anderen Programmierern, die implementierten Funktionen zu nutzen, was ihnen viel Arbeit erspart und andererseits die Bedienung verschiedener Programme vereinheitlicht. Bekanntestes Beispiel dürfte der Dateiauswahlrequester sein, der z. B. von Matt Dillons `Dme`-Editor benutzt wird.

ASH: Die Arp-Shell

Auf den ersten Blick überrascht es, daß die Shell des `ARP`-Projektes, die durch die Datei `L:Ash` repräsentiert wird, geringfügig länger

 © 1990 technicSupport

ist als das `Shell-Seg` der Workbench-Diskette. Dies erklärt sich durch den Umstand, daß die Befehle `=`, `Endif`, `Endskip`, `Lab`, `Alias`, `Endcli`, `Run`, `Else`, `Execute`, `Skip`, `If`, `Quit` und `Failat` bereits in der `ASH` enthalten sind, was die Abarbeitung von Stapeldateien erheblich beschleunigt. Aktiviert werden kann die `ASH` durch den Befehl

```
resident Cli L:Ash system pure add
```

oder durch Aufrufen des `AShell`-Kommandos, das die Shell automatisch lädt, sofern dies nicht schon geschehen ist. Die `Startup-Sequence` einer `ARP`-Diskette kann somit auf die Zeilen

```
AShell from s:Startup
endcli >nil:
```

beschränkt werden. `Startup` ist dann eine Stapeldatei, die alle übrigen Befehle enthält und bereits vom Geschwindigkeitsvorteil residenter Befehle Gebrauch machen kann. Der dazu benötigte Befehl heißt beim `ARP` jedoch nicht mehr `Resident`, sondern `ARes`. Benutzt man den alten Resident-Befehl in der `ASH`, dann wird zwar die Liste residenter Programme wie gehabt verwaltet, die resident geladenen Kommandos werden von der Shell jedoch völlig ignoriert. Man muß also zwangsläufig den `ARes`-Befehl verwenden oder bei der Amiga-Shell bleiben. Dies würde jedoch den Verzicht auf viele weitere Vorteile bedeuten.

`ARes` zum Beispiel geht wesentlich sparsamer mit dem Speicherplatz um und überprüft Befehle automatisch auf ihre Tauglichkeit, während die Workbench 1.3 den Schiedsspruch dem Guru überläßt. Auch lädt `ARes` ein Kommando nicht sofort, sondern beim erstmaligen Aufruf, ein weiteres Verfahren, Speicherplatz zu sparen. Das sofortige Laden kann allerdings mit der Option `Force` erzwungen werden: Auf diese Weise kann man, wenn man vielleicht nur ein Laufwerk besitzt, alle häufig benötigten Befehle schon in der Startup-Sequence laden und so die Zugriffe auf die Workbenchdiskette erheblich reduzieren.

Selbstverständlich unterstützt die `ASH` den `Alias`-Befehl und die Erkennung des Script-Bits. Die Ausgabe von `Alias` läßt sich mit der spitzen Klammer in eine Datei umlenken; Stapeldateien, die über

das Script-Bit aufgerufen werden, akzeptieren sogar die Umlenksymbole > und < in der Befehlszeile: Beides wird von der AmigaShell nicht unterstützt.

`ASH` unterstützt Umgebungsvariable nach dem Manx/Rokicki-Standard, die mit dem Gleichheitszeichen oder dem `Set`-Befehl gesetzt werden können. Interne Variable stehen zum Beispiel für die Nummer des Cli-Prozesses, für den Fehler-Returncode sowie für den `result2`-Fehler des Betriebssystems zur Verfügung.

Pipes vom Feinsten

Interessantestes Feature ist jedoch die Implementation einer internen Pipe, die dem Pipe-Gerät der Workbench weit überlegen ist. Zu ihrer Nutzung muß man allerdings im Besitz des Konsolenhandlers `Conman` (zu finden auf Fishdisk 165) sein. Nach der Aktivierung Conmans — dies empfiehlt sich ohnehin, da `Newcon:` diesem Programm nichts entgegenzusetzen hat — wird das Pipe-Gerät mit dem Befehl

```
mount pip:
```

angemeldet. Die Mountlist muß zuvor um den Eintrag

```
PIP:
Handler = L:ConHandler
Stacksize = 600
Priority = 5
GlobVec = 0
#
```

ergänzt worden sein.

Aufgabe einer Pipe ist es, die Ausgaben eines Befehls ohne Anlegen einer Zwischendatei sofort an ein anderes Kommando weiterzugeben. Durch geschickte Kombination geeigneter Kommandos können sich diese in ihren Fähigkeiten gegenseitig ergänzen. Um etwa ein Verzeichnis nicht nur zu listen, sondern die Ausgabe in alphabetischer und nach Zeilen numerierter Form erfolgen zu lassen, genügt die Eingabe:

```
list | sort | type opt n
```

 © 1990 technicSupport

Mit den ARP-Kommandos (die Commodore-Befehle sind dazu nicht in der Lage) lassen sich sogar Tastatureingaben in eine Pipe leiten. Der Befehl

```
type | sort to ram:sortiert
```

erzeugt eine Datei, in der die einzelnen Eingabezeilen alphabetisch sortiert sind, egal in welcher Reihenfolge sie eingegeben wurden.

Die Erweiterungsbefehle

Die Erweiterungsbefehle, die nicht zum Befehlsumfang der offiziellen Workbench gehören, umfassen sowohl Befehle, die spezielle Eigenschaften der ASH unterstützen, als auch Befehle, die den Standard-Befehlssatz sinnvoll ergänzen.

- ARes ist, wie oben schon besprochen der für die ASH erforderliche Resident-Befehl.
- Set dient zum Setzen der Umgebungsvariablen.
- Cmp vergleicht Dateien miteinander.
- LoadLib veranlaßt das Laden einer Library.
- Move kopiert Dateien und Verzeichnisse unter Löschen der Quelldateien. Im Gegensatz zum Rename-Befehl darf sich das Zielverzeichnis auch auf einem anderen Gerät befinden.
- Read gestattet die interaktive Festlegung von Umgebungsvariablen durch den Benutzer.
- AShell ruft ein Fenster der ASH-Shell auf. Die Befehlssyntax entspricht der des NewShell-Kommandos.
- Basename reduziert eine Umgebungsvariable, die einen kompletten Pfad repräsentiert, auf den reinen Dateinamen.
- TackOn verbindet in umgekehrter Weise einen Pfad und einen Dateinamen zu einer vollständigen Angabe.

- **ARun** dient wie der Run-Befehl zum Aufrufen von Hintergundprozessen, erlaubt jedoch zusätzlich das Setzen von Stackgröße und Priorität. Bei Verwendung der NOIO-Option läßt sich das Fenster, aus dem heraus der Aufruf erfolgte, auch wieder schließen. **ARun** wurde für die Version 1.1 so konzipiert, daß er bereits damals residente Kommandos berücksichtigen konnte. Mit der **ARP**-Shell hat dieser Aspekt seine Bedeutung jedoch verloren.

Fazit

Die Erwartungen, die man an das neue **ARP** stellen konnte, sind mehr als übertroffen worden. Wer nur gelegentlich mit dem AmigaDOS arbeitet, kann sich auf die Verwendung der Standard-Befehle beschränken, um ohne zusätzliche Anstrengungen eine leistungsfähigere Workbench zu erhalten. Anwender mit höheren Ambitionen erhalten mit der **ASH** und den erweiterten Befehlen eine Arbeitsumgebung, die wohl nur noch von der (kommerziellen) **WShell** übertroffen wird — und das quasi zum Nulltarif. Damit keine Irrtümer entstehen: **ARP** ist eine Verbesserung der Workbench 1.3, aber kein vollständiger Ersatz; FastFileSystem, schnelle Druckertreiber usw. gibt es nur auf der offiziellen Workbench-Diskette.

PROBLEME

Die Programmierer haben sich alle Mühe gegeben, ihre Befehle so zu gestalten, daß sie abwärtskompatibel sind zu den Befehlen der offiziellen Workbench 1.3. Bewußt inkompatibel ist allerdings die Verwendung des Sternchens als globales Jokerzeichen, wie es auch auf anderen Systemen wie MS-DOS oder UNIX benutzt wird. Dadurch steht das Sternchen nicht mehr als Symbol für das aktuelle Fenster zur Verfügung, was bei den Befehlen **Copy** und **Type** das Kopieren der Tastatureingaben in eine Datei mit einem Befehl der Art

```
type * ram:Text
```

verhindert. Das Type-Kommando greift jedoch von selbst auf die Tastatur zu, wenn in der Befehlszeile keine Quelldatei angegeben

 © 1990 technicSupport

wird. Auf Wunsch können beide Befehle das Sternchen als Gerätenamen akzeptieren, dazu ist mit

```
set BCPL true
```

die Umgebungsvariable BCPL auf den Wert true zu setzen. In seltenen Fällen kann es auch beim Einsatz des Sternchens als Escape-Symbol zu Konflikten mit der Bedeutung als Jokerzeichen kommen. Die ARP-Entwickler empfehlen daher das Kommando

```
set ESCAPE \
```

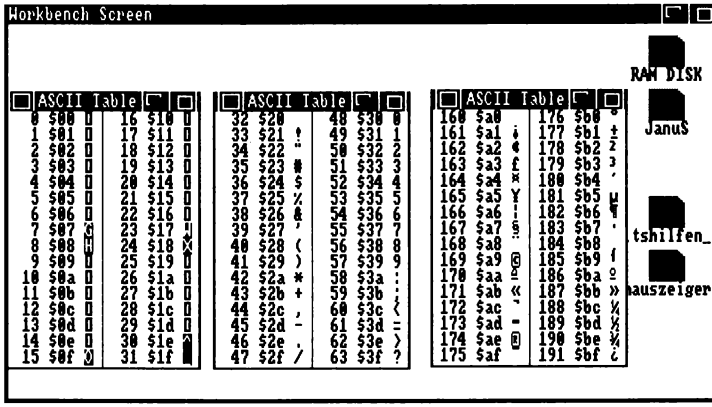
und die anschließende Verwendung des Backslashs als Escape-Zeichen. Weitere Kompatibilitätsprobleme treten selbstverständlich dann auf, wenn man die erweiterten Möglichkeiten der ARP-Befehle nutzt, die von ihren AmigaDOS-Pendants nicht unterstützt werden.

HERKUNFT

ARP 1.3 liegt auf der TBAG-Disk Nr. 31 vor. Die Programmierer behalten sich alle Rechte vor, legen keinen Wert auf finanzielle Unterstützung und freuen sich über jeden Hinweis auf eventuelle Fehler oder Verbesserungsvorschläge und andere Anregungen. Auch bitten sie darum, die Verbreitung des ARP unter den Amiga-Anwendern zu fördern. Die Adresse:

```
AmigaDOS Resource Project
c/o Microsmiths, Inc.
PO Box 561
Cambridge, MA 02140

Electronic Mail:
BIX, People Link: cheath
CIS: 76004,1766
```



Alle ASCII-Codes auf einen Blick

AsciiTable

Anzeige der Tastencodes

AUFRUF

Das Programm wird von der Workbench durch Anklicken des Icons, vom CLI durch Eingabe von

```
asctable
```

gestartet.

ZWECK

Aus unverständlichen Gründen verzichteten bislang viele Amiga-Bücher auf eine Übersicht der verwendeten Tastencodes. Diese Lücke schließt `Ascii-Table`, das in einem Fenster alle Zeichen mit ihren Tastencodes in dezimaler sowie hexadezimaler Form anzeigt.

© 1990 technicSupport

Programmierer können mit diesem Programm schnell den Code für eine bestimmte Taste finden.

BESCHREIBUNG

Nach dem Aufruf des Programmes werden in einem Fenster zweispaltig der dezimale und der hexadezimale Code sowie das zu diesem Wert gehörende Zeichen aufgelistet. Nicht darstellbaren Zeichen ist ein Rechteck zugeordnet. Nach Anklicken dieses Fensters kann man sich mit [↓] und [↑] durch die Tabelle bewegen. Beim Aufruf von `AsciiTable` sollte der Standardzeichensatz eingestellt sein.

HERKUNFT

Das Programm findet sich mit seinem C-Quellcode (Manx 3.40a) auf der RPD 148. Autor ist:

`A. Caleb Gattegno`

AssignDev

Erlaubt die Umbenennung „physikalischer“ Geräte

AUFRUF

`AssignDev` läßt sich ausschließlich aus dem CLI heraus aufrufen. Für die Definition einer neuen Gerätebezeichnung lautet der Befehl:

`AssignDev` ⟨*neues Gerät*⟩: ⟨*existierendes Gerät*⟩:

Mit dem Befehl

`AssignDev` ⟨*neudefiniertes Gerät*⟩:

läßt sich eine getroffene Zuweisung wieder rückgängig machen.

ZWECK

Die Bezeichnungen der „physikalischen“ Geräte, der *Devices*, sind mit dem AmigaDOS-Befehl `assign` nicht zu beeinflussen. Mit `AssignDev` können Laufwerke, Drucker und andere Devices unter jedem gewünschten Namen angesprochen werden.

BESCHREIBUNG

Besitzer eines Amiga 2000 mit einer Festplatte an Stelle des internen Zweitlaufwerkes haben ohne Eingriffe in die Hardware keine einfache Möglichkeit, ein Laufwerk unter der Bezeichnung `df1:` anzusprechen. Dies ist umso ärgerlicher, als viele Anwendungsprogramme wie `DeluxePaint`, `SONIX`, Textverarbeitungen usw. in ihren Requestern keinen [DF2:] -Button anbieten. Der Zugriff auf das (externe) Zweitlaufwerk erfordert jedesmal die Eingabe der Gerätebezeichnung per Hand.

Versucht man, diese Unbequemlichkeit mit Hilfe des `assign`-Befehls abzustellen, kann man mehrere Enttäuschungen erleben. Gibt man den Befehl

```
assign df1: df2:
```

 © 1990 technicSupport

ein, wenn sich keine Diskette im Zweitlauf befindet, so erscheint zunächt ein Requester mit der Fehlermeldung

```
No disk present in unit 2
```

und anschließend — nach dem Wegklicken des Requesters — die lapidare Mitteilung:

```
Can't find df2:
```

Legen wir also eine Diskette ein und versuchen es nochmal. Irgendwie werden wir die „Kiste“ schon gefügig machen. Das Prompt erscheint, na also! Wir machen ernst:

```
dir df1:
```

Tatsächlich wird der Inhalt der Diskette im Laufwerk `df2:` ausgegeben. Wir dürfen Amiga wieder liebhaben. Wie lange wird seine Zuneigung anhalten? Wir sind ja so undankbar, daß wir ihm eine Diskette (die in `df2:`) wegnehmen. Er bekommt dafür aber eine neue:

```
dir df1:
```

Ende des lieben Friedens! Amiga will die alte Diskette wiederhaben. Was hat ihn denn nun schon wieder verstimmt? Der `assign`-Befehl, ohne Parameter eingegeben, klärt die Sache auf. Amiga hat „df1“ als logische Gerätebezeichnung verstanden und ihr das Basisverzeichnis der seinerzeit in `df2:` befindlichen Diskette zugeteilt. Daher findet sich „df1“ auch unter den Bezeichnungen der Directories, nicht aber im unteren Teil der Liste bei den Devices. Klar, daß er nach Eingabe unseres `dir`-Befehls diese Diskette braucht, um das Verzeichnis auszugeben. Fehlschläge auf der ganzen Linie!

Wir werden uns dennoch durchsetzen! Entweder wir definieren uns ein `df1:`-Laufwerk in der *Mountlist* oder — viel einfacher und flexibler — wir verwenden das Public-Domain-Programm `AssignDev`. Wir kopieren es in das `c`-Verzeichnis unserer Workbench und versuchen es mit

```
AssignDev df1: df2:
assign
```

Und tatsächlich erblicken wir unter den aufgeführten Devices das lang ersehnte `DF1`. Amiga hat die „Kröte" geschluckt. Nun läßt sich sowohl über „df1:", als auch über „df2:" auf das Zweitlaufwerk zugreifen. Auch ein Diskettenwechsel ist jetzt erlaubt. `AssignDev` kann noch mehr, zum Beispiel `Prt:` in `Drucker:` umbenennen. Hierüber ließe sich — zusammen mit dem `assign`-Befehl — eine zwar fragwürdige, aber doch recht weitgehende Eindeutschung des AmigaDOS durchführen. Zuweisungen lassen sich jederzeit wieder rückgängig machen. So befreit

```
AssignDev df1:
```

unser Zweitlaufwerk von seiner zeitweiligen Schizophrenie. Physikalische Geräte wie `prt:` lassen sich auf diese Weise natürlich nicht beseitigen.

BEMERKUNGEN

Wer eine Janus-Festplatte `jh0:` auch als `dh0:` ansprechen möchte, kann in der Regel den normalen `assign`-Befehl verwenden, da sich eine Festplatte ja nicht aus ihrem Laufwerk nehmen läßt. Es spielt dann keine Rolle, ob „dh0:" ein Verzeichnis oder ein Gerät bezeichnet — es sei denn, Sie geben Ihrer Festplatte laufend neue Namen.

PROBLEME

Auch wenn Sie das Zweitlaufwerk als `df1:` ansprechen, beziehen sich alle Fehlermeldungen nach wie vor auf `df2:`. Dies kann anfangs vielleicht etwas Verwirrung stiften.

Ernsthaftere Schwierigkeiten kann es mit Programmen geben, die die Laufwerke direkt ansprechen. Das Kopierprogramm `TurboBackUp` zum Beispiel läßt sich nicht starten, wenn einem der angeschlossenen Laufwerke mit `AssignDev` ein weiterer Name zugewiesen worden ist.

HERKUNFT

Das Programm stammt von PHILLIP LINDSAY. OLIVER SEIBERT hat es modifiziert und einige Bugs beseitigt. Diese verbesserte Version findet man einschließlich des Sourcecodes auf Fishdisk 111.

© 1990 technicSupport

Study Aid Card File

Ein elektronischer Karteikasten

AUFRUF

Von der Workbench aus wird das Programm durch Anklicken des `Card File`-Icons gestartet.

ZWECK

`Study Aid Card File` ist ein elektronischer Zettelkasten. Das Programm verwaltet Kurztexte, wie sie auf üblichen Karteikarten stehen würden. Da Vorder- und Rückseiten unterstützt werden, kann `Study Aid Card File` als Lernhilfe oder auch für Frage- und Antwortspiele nach der Art von *Trivial Pursuit* eingesetzt werden.

BESCHREIBUNG

Installation

Vor Gebrauch des Programms sollten die entsprechenden Programmdateien auf eine eigene Diskette kopiert werden. Da `Card File` in AmigaBasic geschrieben ist, ist bei der Installation auf folgendes zu achten:

1. Bei Programmstart wird automatisch auf `AmigaBasic` zugegriffen. Es ist daher nötig, entweder das `AmigaBasic` von der `Extras`-Diskette auf die neue `Card File`-Diskette (und zwar in das Hauptverzeichnis) zu kopieren, oder aber dem Programm mitzuteilen, wo es `AmigaBasic` finden kann. Dazu wird das `Card File`-Icon auf der Workbench einmal (!) angeklickt. Anschließend wird auf der Workbench das Programm `Info` (zu finden im `Workbench`-Menü) aufgerufen. Unter `Default Tool` wird der Pfadname eingegeben, unter dem `AmigaBasic` zu finden ist. Bei den meisten Disketten müßte der Eintrag „`ExtrasD:amigabasic`“ lauten, andernfalls ist der Diskettenname „`ExtrasD`“ entsprechend abzuändern. Schließlich wird das Info-Programm mit [Save] verlassen.

© 1990 technicSupport

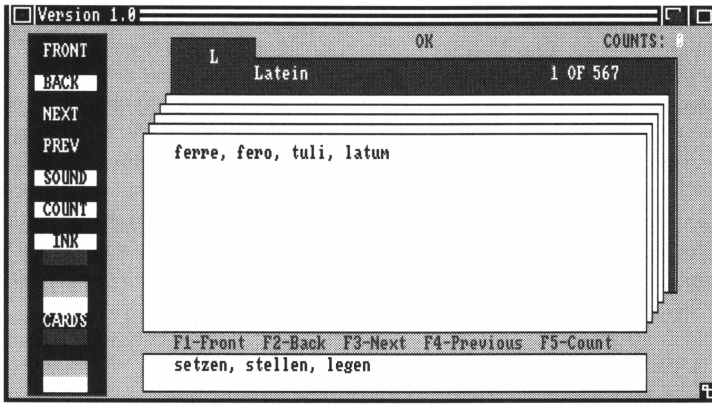
2. Das Programm benutzt zur Verwaltung der Information ein Indexfile, also eine Datei, die Zeiger zum Ordnen der Textsammlung und Eintragsnummern enthält. Diese Indexdatei heißt, wer hätte das gedacht, `CARDFILES` und es ist wichtig, daß das Programm beim Aufruf diese Datei findet. Daher sollte — in der oben beschriebenen Weise — in der „.info"-Datei der Pfadname zu `CARDFILES` angegeben werden. Diesmal wird der Pfad jedoch unter `Tool Types` eingetragen und mit [Save] abgespeichert.

3. Das Programm besitzt ein `Defaults`-Menü, das weiter unten beschrieben wird. In diesem Menü kann der Benutzer eigene Vorlieben bezüglich der Farbeinstellungen, der Menüführung usw. vermerken, so daß sie beim nächsten Programmstart automatisch berücksichtigt werden. Auch der Pfad oder das Laufwerk, in dem eigene Textdateien abgespeichert werden, müssen in der „Defaults"-Datei vermerkt werden. Sobald der Benutzer Veränderungen vornimmt, wird diese Datei angelegt oder verändert. Um beim nächsten Programmstart diese Änderungen zu berücksichtigen, muß das Programm die `Defaults`-Datei finden können. Dazu wird wieder der entsprechende Pfadname als `Tool Type` in das Info-Menü eingetragen und abgespeichert.

Grundsätzliches zum Arbeiten mit Card File

`Card File` wurde mit der Absicht konzipiert und programmiert, eine Lernhilfe für Schule und Studium zu schaffen. Der Programmierer schreibt in der ausführlichen Dokumentation, wie er in der Schule immer mit Karteikarten gearbeitet hat, um z. B. die Vokabeln einer Fremdsprache zu lernen. Auf der Vorderseite einer solchen Karte stand dabei der Ausgangsbegriff und auf der Rückseite das gesuchte Äquivalent in der Fremdsprache. So eingesetzt, bietet `Card File` in der Tat eine ausgezeichnete Lernhilfe, beispielsweise für Lateinvokabeln. Dies könnte folgendermaßen aussehen: Es wird eine Datei namens `Latein` angelegt. Als Unterordnung zum Fach Latein kann den Karten jeweils ein spezielleres Sachgebiet zugeordnet werden. Angenommen, wir wollen für das Fach Latein drei Sachgebiete definieren: 1) Vokabeln, 2) Stammformen und 3) Deklinations-

 © 1990 technicSupport



CardFile verwaltet Daten nach dem Prinzip eines Karteikastens

und Konjugationsendungen. Innerhalb der angelegten Datei `Latein` kann durch Zuordnung der Karten zu diesen drei Gebieten ganz systematisch gelernt werden. Fangen wir mit dem Vokabeltraining an: Es wird eine Karte unter dem Sachgebiet „Vokabeln“ angelegt. Wie das genau geht, wird weiter unten bei der Beschreibung der Menüs vorgeführt. Auf der Vorderseite der Karte wird das deutsche Ausgangswort eingetragen, zum Beispiel „setzen, stellen, legen“; auf der Rückseite der Karteikarte steht dann das lateinische Äquivalent: „ponere“. Hier haben wir schon die Möglichkeit, direkt die Stammformen oder auch andere Informationen wie Personalendungen oder Art der entsprechenden Konjugation einzugeben, denn auf der Kartenrückseite können bis zu 550 Zeichen in maximal 11 Zeilen eingetragen werden. Auf diese Weise legen wir für jede Vokabel ein Kärtchen an. Ist die Datei komplett eingegeben, kann die Vokabelabfrage gestartet werden, indem nämlich die Karte von vorne betrachtet und das lateinische Wort gesucht wird. Die Antwort wird durch Ansehen der Kartenrückseite kontrolliert. Dann wird die nächste Vokabel gepaukt...

Stammformen können ebenso leicht gelernt werden. Dazu wird auf der Vorderseite der Infinitiv Präsens Aktiv, z. B. „ferre“ eingetragen. Auf der Rückseite der Karte könnten dann zunächst die Übersetzung, „tragen“, sowie die Stammformen „fero - tuli - latum“ stehen. Diese Karten würden dann unter dem Sachgebiet „Stammformen“ erstellt und abgespeichert werden. Unter dem Sachgebiet „Konjugationen/Deklinationen“ abgespeicherte Karten können zum Erlernen von Kasus- und Personalendungen eingesetzt werden. Auf der Kartenvorderseite würde dann z. B. „Konsonantische Deklination“ stehen, und auf der Rückseite könnten die verschiedenen Endungen stehen.

Die Bedienung des Programms

Nach der Beschreibung des Prinzips von `Card File` nun zur Bedienung des Programms: Wenn das Programm richtig installiert ist, und es durch Anklicken des Icons aufgerufen wird, erscheint zunächst ein Fenster, in dem graphisch eine Reihe von übereinanderliegenden, rechteckigen Karteikarten dargestellt wird. Wer in gewohnter Amiga-Manier die rechte Maustaste gedrückt hält, stellt fest, daß der Programmierer die Mausbedienung des Amiga voll unterstützt hat. Es gibt vier Pull-Down-Menüs: `Project`, `Cards`, `Defaults` und `Search`.

Das Project-Menü

`New` Durch Anwählen von `New` wird eine neue Karteikarten-Datei angelegt. Der Benutzer vergibt zunächst einen Dateinamen, wobei das Programm automatisch prüft, ob eine Datei mit diesem Namen bereits existiert.

`Open` Soll eine Datei benutzt werden, die bereits vorher angelegt und auf der Diskette abgespeichert worden ist, kann diese mit dem `Open`-Befehl geöffnet werden. Es erscheint ein Fenster, in dem alle verfügbaren Karteikarten-Dateien mit einer Kennziffer verzeichnet sind. Diese Kennziffer wird per Tastatur eingegeben und die entsprechende Datei steht zur Bearbeitung oder zum Lernen zur Verfügung.

 © 1990 technicSupport

`Quit` Der `Quit`-Befehl tut genau das, was alle `Quit`-Befehle tun, er führt nämlich zum Verlassen des Programms. Ist man auf die Workbench-Ebene zurückgekehrt, sieht man, daß das Programm während der Arbeit selbständig eine RAM-Disk eingerichtet und verwaltet hat. Das ist ein kleiner Nachteil für alle diejenigen Amiga-Benutzer, für die Speicherplatz ein wertvolles Gut ist, denn auch wenn diese RAM-Disk leer ist, nimmt sie doch Speicherplatz in Anspruch, der beim besten Willen vor dem nächsten Reset nicht wieder geräumt wird.

`Delete` Dieser-Befehl löscht eine Datei, die der Benutzer bestimmen kann. Dabei ist es gleichgültig, ob diese Datei bereits geöffnet wurde oder nicht. Genaugenommen wird nicht die Datei gelöscht, sondern die in der `CARDFILES`-Datei angelegten Zeiger.

Das Cards-Menü

Dieses Menü verwaltet mit seinen Befehlen `Add`, `Delete` und `Print` die einzelnen Karteikarten. Leider ist der Editor, der zum Erstellen der Kartentexte benutzt werden muß, sehr spartanisch und unkomfortabel geraten. Die einzige Editiermöglichkeit besteht darin, mit [Backspace] falsche Eingaben zu löschen. Einmal eingerichtete Karten sind im Text leider nicht mehr veränderbar, ebensowenig gibt es eine Möglichkeit, andere Editoren einzubinden.

`Add` Mit dem `Add`-Befehl wird eine neue Karte eingerichtet. Das Programm öffnet ein Fenster, in das der Text eingetragen wird. Zunächst wird das Sachgebiet bestimmt: In der oben beschriebenen Anwendung als Lateinvokabel-Trainer wäre hier z.B. „Vokabeln“ einzutragen. Dann wird die Vorderseite der Karte beschrieben. Hierfür stehen nur zwei Zeilen zu je 50 Zeichen zu Verfügung. Dann wird der Eintrag für die Kartenrückseite vorgenommen: Hierfür stehen schon 11 Zeilen zu je 50 Zeichen bereit. Der Editor zentriert die Einträge automatisch. Die einzige Möglichkeit, Einträge linksbündig zu gestalten, ist das Auffüllen der Zeile mit Leerzeichen.

`Delete` Der `Delete`-Befehl löscht die aktuelle Karteikarte aus der Datei und die Zahl der angelegten Karten wird um 1 kleiner.

`Print` Der `Print`-Befehl schließlich aktiviert ein kleines Druckprogramm, das den Karteninhalt ausdrucken kann, und zwar unter Benutzung der in den `Preferences` eingestellten Druckerparameter und -treiber.

Das Defaults-Menü

Wie bereits beschrieben, lassen sich von Benutzerseite her Änderungen am Erscheinungsbild des Programms vornehmen, die bei jedem Neustart automatisch zur Verfügung gestellt werden sollten. Das `Defaults`-Menü enthält die dazu benötigten Parameter. Es wurde eine sogenannte *Gadget Box* programmiert, die eine individuelle Einstellung der Hintergrundfarbe der Karten, der Schriftfarbe und anderer Parameter ermöglicht.

`Show` Aktiviert die Gadget Box.

`Hide` Blendet die Box wieder aus.

`Change` Der `Change`-Befehl bietet drei Optionen: `Autoload` wird entweder aktiviert oder bleibt ausgeschaltet. Ein aktiviertes Autoload bewirkt, daß eine Datei beim Programmstart automatisch geladen wird. Ist die zweite Option namens `Autosave` eingeschaltet, wird die aktuelle Karteikarte einer weiteren Datei zugeordnet, deren Name dem Sachgebiet der Karte entspricht. In oben beschriebener Anwendung würde also neben der Datei „Latein“ eine zweite Datei „Vokabeln“ angelegt. Der Benutzer kann dann dem Programm mitteilen, ob eine Vokabel, die er nicht gewußt hat, in diese zweite Datei kopiert werden soll. Alle nicht gewußten Vokabeln können so in der zweiten Datei gesammelt und noch einmal gepaukt werden. Die dritte Option ist besonders wichtig, wenn man `Card File` von einem anderen Laufwerk als dem eingebauten Laufwerk `df0:` starten möchte. Unter `Data Drive` wird dann das externe Laufwerk `df1:`, `df2:` usw. oder die Festplatte `dh0:` bzw. `jh0:`

 © 1990 technicSupport

eingegeben, auf denen `Card File` die Dateien suchen und abspeichern wird.

`Save` Mit `Save` werden die geänderten Angaben in der „Defaults"-Datei gespeichert.

Das Search-Menü

`By Topic` Ähnlich wie bei einer relationalen Datenbank oder einer Textverarbeitung lassen sich bestimmte Einträge auch suchen, zum Beispiel nach Sachgebiet (`By Topic`). Stehen in unserer „Latein"-Datei Karten zu Vokabeln, Stammformen und Konjugationen in ungeordneter Reihenfolge nacheinander, können wir das Programm z. B. veranlassen, von einer Vokabelkarte direkt zur nächsten zu springen.

`By Pattern` `By Pattern` durchsucht alle Vorderseiten nach bestimmten Textstrings. Auf diese Weise kann `Card File` auch als kleines Wörterbuchprogramm eingesetzt werden. Einfach das Suchwort als Textstring eingeben und auf der entsprechenden Karte das Äquivalent der Zielsprache ablesen.

`By Card Number`
Die letzte Option des `Search`-Menüs schließlich funktioniert wie eine GOTO-Funktion. Will man von Karte Nummer 4 zu Karte 78 springen, gibt man dem Programm lediglich vor, nach Karte Nummer 78 zu suchen, und es springt automatisch an die entsprechende Stelle der Datei.

BEMERKUNGEN

Obwohl `Card File` ja eigentlich als Lernhilfe zum Pauken abfragbaren Wissens gedacht und konzipiert ist, lassen sich auch die angenehmeren Dinge des Lebens mit dieser PD-Software verwalten. Alles, was sonst in einem Zettelkasten Platz findet, kann man nämlich auch hier unterbringen. So zum Beispiel Zitate: GEORGE BERNARD

SHAWS berühmter Spruch „Das Einzige, was die Engländer von den Amerikanern unterscheidet, ist ihre gemeinsame Sprache“ ließe sich so zu einer ganzen Sammlung von Weisheiten ausbauen, die mit `Card File` verwaltet werden.

Auch das mittlerweile sehr beliebte *Trivial Pursuit* bietet sich als klassisches Frage- und Antwortspiel geradezu für `Card File` an. Der einzige Nachteil beim Verwalten von Trivial Pursuit-ähnlichen Fragekarten ist, daß aufgrund des beschränkten Platzes auf der Vorderseite der `Card File`-Karten immer nur eine Frage stehen kann.

Ich selbst benutze `Card File` seit einiger Zeit zum Verwalten meiner privaten Bibliothek: Für jedes Buch in meinem Regal ist ein passendes Kärtchen unter `Card File` angelegt, auf dem Autor, Titel usw. verzeichnet sind. Zur Verwaltung von Audio- und Videocassetten wäre das Programm ebenfalls gut geeignet.

Wer `Card File` wirklich für die Schule und das Studium nutzt, darf natürlich nicht vergessen, daß sich nur eine bestimmte Art von Wissen auf diese Weise lernen läßt: Hier geht es um reines abfragbares Pauken. Das Programm fördert in keiner Weise andere Fähigkeiten und Fertigkeiten, wie sie von Schülern, Studenten und anderen Lernenden erwartet werden: logisches Denkvermögen, das Erkennen von Zusammenhängen oder Kreativität. Aber es darf auch nicht vergessen werden, daß es einfach mehr Laune macht, mit dem Amiga Vokabeln zu lernen, als auf herkömmliche Weise!

Am Programm selbst sind eigentlich nur zwei Dinge zu kritisieren: Zum einen die lausigen Editiermöglichkeiten der Kartentexte — der Autor verspricht allerdings in zukünftigen Versionen Abhilfe! — und zum anderen die fehlende Möglichkeit, verschiedene Dateien in relationaler Weise zu verknüpfen. Relationale Verknüpfungen, wie bei einem Datenbankprogramm, würden es in diesem Falle erlauben, gewisse Vokabeln in mehreren Sprachen anzusprechen, also eine Deutsch-Latein- und eine Deutsch-Französisch-Datei gleichzeitig geöffnet zu haben und den deutschen Begriff gleichzeitig in mehreren Sprachen zu lernen. Hier bietet sich die Chance für unsere Leser, das Basic-Programm entsprechend zu erweitern und als Public Domain zur Verfügung zu stellen...

 © 1990 technicSupport

HERKUNFT

Die hier besprochene Version 1.0 befindet sich auf der Fishdisk 090. `Study Aid Card File` wurde von JAMES B. PARKS geschrieben. Es ist als Shareware deklariert, der Autor erwartet vom Benutzer eine Gebühr von $15 an die folgende Adresse:

```
James B. Parks
5450 W 102nd Place
Broomfield
Colorado 80020
USA
```

Chess

Ein Schachprogramm in zwei Entwicklungsstadien

VORBEMERKUNG

Das im folgenden vorgestellte Schachprogramme begnügte sich mit einer sehr spartanischen Benutzerführung, bis sich ein weiterer Programmierer des mitgelieferten Quellcodes annahm und das Programm mit einer grafischen Oberfläche ausstattete, wobei er den Programmkern unverändert ließ. Dies ereignete sich zwischen dem Erscheinen des zweiten und des vierten Bandes unserer Public Domain Reihe.

Da es recht interessant ist diese Entwicklung nachzuvollziehen, haben wir bei dieser Programmbeschreibung darauf verzichtet, den ursprünglichen Artikel aus Band II und die Update-Informationen aus Band IV zusammenzufassen. Da beide Beschreibungen ganz verschiedene Aspekte des Programms betonen, ergeben sich durch diese Darstellungsform auch keinerlei Verständnisschwierigkeiten.

AUFRUF

Das Programm wird vom CLI aus durch Eingabe von

```
Chess
```

aufgerufen, von der Workbench aus durch Anklicken des `Chess`-Icons.

BESCHREIBUNG

Schach — das Spiel der Könige als Public-Domain-Programm auf Amiga! Um es gleich vorweg zu nehmen: Dieses Programm ist bezüglich der Spielstärke den zur Zeit verfügbaren kommerziellen Schachprogrammen zwar nicht ganz ebenbürtig, aber doch vergleichbar. Wer nicht gerade Bundesliga-Niveau erreicht, für den ist `Chess` ein durchaus ernstzunehmender Schachpartner. So positiv die spieltechnische Programmierung ausgefallen ist, so negativ fällt die graphische Umsetzung ins Gewicht, sie wird in keiner Weise den

© 1990 technicSupport

Möglichkeiten des Amiga gerecht. Agnus, Paula und Denise können ruhig Urlaub machen! Die Darstellung des Schachbretts und der Figuren auf dem Amiga sind durch simple ASCII-Zeichen realisiert, so daß man nicht umhin kommt, sich ein richtiges Schachbrett neben den Rechner zu stellen. Auch die Maus kann Ferien machen: die Züge sind durch Tastaturkommandos (z.B. [e] [2] [e] [4] = e2e4) einzugeben.

Doch nun wieder eine gute Nachricht: Neben dem lauffähigen Programm ist auf Fishdisk 096 auch der Quellcode in „C“ verfügbar. Dies bietet nicht nur faszinierende Einblicke in die programmtechnische Realisierung eines Schachprogramms, sondern auch die Möglichkeit, das Programm graphisch etwas aufzubereiten.

Die Schachprogrammierung von „Chess“

Grundsätzlich muß ein Schachprogramm in der Lage sein, erstens die Menge aller möglichen Züge zu bilden und zweitens, Stellungen zu bewerten. Problematisch ist dabei, daß ein guter Schachspieler viele Züge im voraus berechnen muß. Aus der Anzahl der möglichen Züge potenziert sich aber die Gesamtzahl der möglichen Stellungen, je weiter der Spieler vorausplant. Müßte der Rechner alle diese Stellungen bewerten, bräuchte er sehr lange. Daher sind solche sogenannten reinen *brute force*-Programme, die alle möglichen Stellungen bis zu einer bestimmten Tiefe bewerten, wenig effizient. Die Güte eines Programms liegt vielmehr darin, die Verfolgung sinnloser oder offensichtlich schwacher Züge einzustellen, um so die wertvolle Rechenzeit für die Analyse vielversprechender Stellungen zu nutzen. Der rein quantitativen wird so eine qualitative Methode beigefügt. Das vorliegende `Chess`-Programm benutzt hierzu die *alpha-beta*-Methode. Es würde hier zu weit führen, diese im Detail zu beschreiben, doch C-Programmierer haben ja die Möglichkeit, die Realisierung einer solchen alpha-beta-Strategie im Quellcode nachzuvollziehen oder ggf. sogar zu verbessern. Die Unterprogramme `search()` und `CapturSearch()` können als Ansatzpunkte zur Verbesserung der heuristischen Strategie dienen. Das Programm `ScorePosition()` bewerkstelligt die Stellungsanalyse, wobei jeder noch vorhandenen Figur ein Wert zugemessen wird, der sich u.a.

© 1990 technicSupport

auch aus ihrer Stellung auf dem Brett ergibt. Isolierte Bauern z.B. oder Springer am Rand werden nur sehr schwach bewertet, während Zentralfiguren mit Bedrohungen gegnerischer Figuren mehr wert sind.

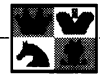
Die Schwäche von `Chess` dürfte in der Eröffnung liegen. Zwar ist in der Datei „chess.opn“ eine Eröffnungsbibliothek implementiert, doch diese schien im praktischen Vergleich nicht umfangreich genug zu sein, so daß `Chess` bereits in der Eröffnungsphase gegenüber anderen Programmen an Boden verlor. Das Mittelspiel ist bei den meisten Programmen recht spielstark, denn dort kommt es nicht so sehr auf *tiefe*, sondern auf *breite* Suche an. Hier macht `Chess` keine Ausnahme. Im Endspiel war `Chess` erstaunlich stark, obwohl hier bei Schachprogrammen allgemein Schwächen liegen, da in dieser Spielphase oft viele Züge voraus geplant werden muß, woraus eine riesige Anzahl möglicher Stellungen resultiert. Zudem haben erfahrene Schachspieler für bestimmte Endkonstellationen, wie „König und Läufer gegen König“, Taktiken parat, die `Chess` nicht kennt. Trotzdem sei nochmals betont, daß `Chess` für den durchschnittlichen Schachspieler ein ernstzunehmender Gegner ist.

Zur Bedienung von „Chess“

Nach dem Programmaufruf wird die Spielstärke des Rechners festgelegt: Der Spieler wird aufgefordert, einen Wert in Sekunden einzugeben, der angibt, wieviel Bedenkzeit dem Amiga zusteht. Dann wartet das Programm auf Tastatureingaben. Es kann ein Spielzug in der Syntax „`e2e4`“, „`b1c3`“ usw. eingegeben werden, oder einer der folgenden Befehle:

`help`	Ruft ein Hilfsmenü auf.
`set`	Erlaubt das Setzen von Figuren. Syntax: „`pe4`“ für „Bauer auf e4“.
`switch`	Spieler und Computer wechseln die Figuren.
`go`	Passen
`reverse`	Das Brett wird umgedreht.

© 1990 technicSupport



`both`	Amiga spielt gegen Amiga.
`time`	Verändert die Spielstärke.
`post`	Amiga macht Strategievorschlag.
`hint`	Zugvorschlag
`list`	Notation der Partie wird als Datei „chess.lst“ abgelegt.
`save`	Speichern der aktuellen Stellung
`get`	Laden einer abgespeicherten Stellung
`quit`	Verlassen des Programms

Rochaden sind ebenfalls möglich: „`0 - 0`“ führt die kleine, „`0 - 0 - 0`“ führt die große Rochade aus. Das Programm hat auch eine *en-passant*-Funktion implementiert. Eine dreimalige Zugwiederholung wird als Remis erkannt. Was fehlt, ist lediglich eine Zugrücknahmefunktion.

Nun zum Programm selbst: Wie bereits erwähnt, enttäuscht die „Graphik“ sehr. ASCII-Striche stellen die Felder und ASCII-Buchstaben die Figuren dar. Aus diesem Grunde muß man auch die englische Bezeichnung der Schachfiguren kennen:

`p` *Pawn* — Bauer

`n` *Knight* — Springer

`b` *Bishop* — Läufer

`r` *Rook* — Turm

`q` *Queen* — Dame

`k` *King* — König

Hier ergibt sich der erste Ansatz für C-Programmierer: Beim Source-Code können im Unterprogramm `SetBoard()` die Buchstabenkürzel entsprechend den deutschen Abkürzungen umgeschrieben werden. Die sechs `if`-Abfragen sind unübersehbar: Statt „`if (s[0]=='p')`“ einfach „`if (s[0]=='b')`“ schreiben, und der Bauer ist kein pawn mehr, sondern ein Bauer. Und hier ein zweiter Tip: Mit einem Font-Editor könnten kleine Schachfiguren als Fonts definiert werden, ein

Satz Figuren in hell auf dunklem Hintergrund für Weiß und ein Satz in dunkel auf hell für Schwarz. Für das Schachbrett könnten ebenfalls entsprechende Strichmuster als ASCII-Zeichen definiert und abgelegt werden. Im Unterprogramm `PrintBoard()`, wo das Brettmuster definiert wird, müßten die entsprechenden Zeichen ersetzt werden. Der abgewandelte Source-Code ist anschließend neu zu kompilieren (z.B. mit `Manx Aztec 3.20a`), um eine geänderte, lauffähige Version zu erhalten. Vor deren Aufruf braucht lediglich der mit dem Font-Editor erzeugte Zeichensatz geladen werden, und schon dürfte das Programm optisch attraktiver wirken.

HERKUNFT

Das lauffähige `Chess`-Programm ist mit seinem Source-Code in C auf Fishdisk 096 zu finden. Der Autor ist John Stanbeck. Für Amiga wurde es umgeschrieben von:

```
Bob Leivian
2702 W. Curry
Chandler AZ 85224
U.S.A.
```

Der Autor lädt alle Amiga-Benutzer ein, sein Programm — vor allem im graphischen Bereich — zu verbessern.

VERWEISE

Mehrere kommerzielle Programme werden als Amiga-Schachpartner angeboten (Chessmaster, Sargon usw.). Auf dem Public-Domain-Sektor dürfte `Chess` noch konkurrenzlos sein.

DIE NEUE VERSION

Vielleicht war es gerade unsere Anregung aus dem zweiten Band unserer PD-Reihe, die die Verbesserungen in der neuen Version 2.0 inspiriert hat, wir wissen es nicht. Tatsache ist, daß manche Schwachpunkte der älteren Version, besonders die fehlende grafische Umsetzung, in der nun vorliegenden neuen Version erheblich verbessert wurden. Früher mußte man bei `Chess` mit einem Schachbrett aus Strichen und Buchstaben vorlieb nehmen, und auch die Züge

© 1990 technicSupport

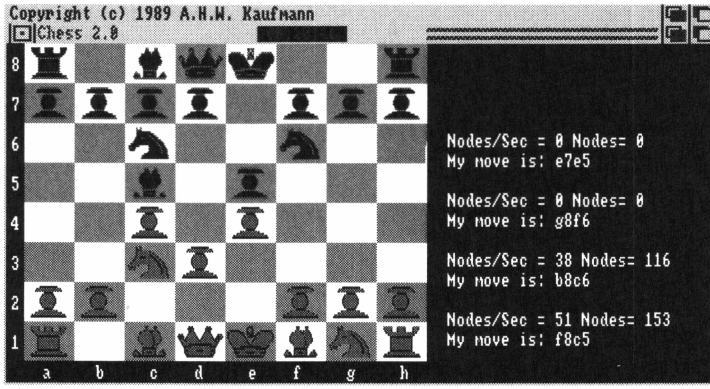
mußten per Tastatur recht mühsam eingegeben werden. Diese Art der Bildschirmoberfläche und der Benutzerführung wurde natürlich in keiner Weise den Amiga-Möglichkeiten gerecht. Es schien eigentlich nur eine Frage der Zeit zu sein, bis ein findiger Programmierer das Programm um eine grafische Oberfläche erweiterte, zumal der Quellcode in C damals dem eigentlichen Programm beilag.

A. H. W. KAUFMANN hat diese Basis nun aufgenommen und sie mit einer für Amiga-Verhältnisse akzeptablen Grafik-Oberfläche versehen, die sich durchaus sehen lassen kann. Pull-Down-Menüs lassen Befehle aufrufen, und die Figuren werden nunmehr mit der Maus bewegt. An der Spielstärke hat sich dadurch natürlich nichts geändert. Wir hatten an der alten Version bemängelt, daß Schwächen vor allem in der nicht sehr umfangreichen Eröffnungsbibliothek lagen, wodurch es leicht möglich war, das Programm durch unkonventionelle Eröffnungszüge in den Nachteil zu bringen. Im Mittelspiel, wo bekanntlich sehr viele Möglichkeiten für legale Züge zu bewerten sind, ist `Chess` nach wie vor ein ernstzunehmender Partner, und auch im Endspiel ist das Programm brauchbar.

In der neuen Version öffnet `Chess` einen neuen Screen mit schwarzem Hintergrund. Die Eingabe der Bedenkzeit, die wir dem Rechner bei seinen Zügen überlassen wollen, geschieht nun in einer Requester-Box, die sich automatisch öffnet. Das Schachbrett mit den Figuren nimmt links oben den größten Teil des Bildschirms in Anspruch. Der freie Platz rechts daneben ist für die Anzeige der Züge und für Systemmeldungen bestimmt.

Eine Beschreibung des Schachbretts erübrigt sich an dieser Stelle. Die Figuren sind gut erkennbar, wenn auch keine Kunstwerke. Eine 3D-Funktion, wie in anderen Schachprogrammen fast üblich, gibt es hier nicht; alles ist auf Funktionalität ausgerichtet. Wir hatten bei unseren ersten Versuchen, die Figuren zu bewegen, etwas Schwierigkeiten; sicherlich ist dies gewöhnungsbedürftig. Alles wird über die linke Maustaste gesteuert. Die zu bewegende Figur wird einmal angeklickt, woraufhin der Rand des Feldes zweimal rot aufblinkt. Danach wird das Zielfeld genauso angeklickt. Unter heftigsten Blinkbewegungen wandert die Figur dann ins Zielfeld. Ein Zug, der nicht den Regeln entspricht, wird natürlich abgefangen und angezeigt.

© 1990 technicSupport



Das Schachprogramm in seinem neuen Gewand

Die Rochaden sind auszuführen, indem der König bewegt wird. Der Turm zieht dann automatisch nach.

Bei der laufenden Zeiteinblendung gibt es einige Kritikpunkte. Wenn `Your Move` angezeigt wird, ist der Spieler am Zug. Ist der Amiga dran, erscheint die Anzeige `Elasped Time`, was natürlich nur ein Schreibfehler ist und eigentlich `Elapsed Time`, also verstrichene Zeit, heißen müßte. Der Amiga nimmt unabhängig von der Komplexität des nächsten Zuges die Bedenkzeit wahr, die der Spieler ihm zugestanden hat, d. h. selbst wenn ein Zug ganz offensichtlich der sinnvollste Zug ist, rechnet er weitere mögliche Züge durch. Schließlich ist zu kritisieren, daß die Zuganzahl die Gesamtzüge angibt und nicht, wie in der Schachnotation üblich, die Zahl der Züge pro Spieler.

Zu den Menüs: Im `Project`-Menü stehen mit `Load Game` und `Save Game` Befehle zur Verfügung, die das Abspeichern und Laden von Spielen oder Spielständen ermöglichen. Mit dem Befehl `Set Board` können Figuren einzeln und individuell auf dem Spielfeld plaziert

© 1990 technicSupport

werden. `List Moves` generiert das Listing aller bisher gemachten Spielzüge und legt es in einer Datei ab. Der `Colors`-Befehl erlaubt eine individuelle Farbeinstellung der Bildschirmoberfläche. Was an fehlenden Grafikmöglichkeiten in der alten Version zu kritisieren war, ist hiermit also wesentlich besser gelöst. Diese `Colors`-Möglichkeit ist im übrigen von einem anderen Programmierer geschrieben worden als A. H. Kaufmann. Wer sich das Credits-Menü ansieht, wird merken, daß die vorliegende Programmversion durch sehr viele Programmiererhände gegangen ist. Das ist das Schöne an einem Amiga-PD-Programm, bei dem der Quellcode beiliegt: Jedem ist es möglich, das Programm weiter zu verbessern, so daß ein Teamwork von Leuten entsteht, die sich gar nicht kennen. Mit `Read Docs` im Project-Menü können die Erläuterungstexte gelesen werden, und mit `Quit` wird das Programm verlassen.

Im `Play Chess`-Menü steht zunächst der Befehl `Switch Sides`, bei dem Spieler und Computer die Figuren tauschen, zur Verfügung. Dies ist auch noch mitten im Spiel möglich. Mit `Reverse Board` behalten Spieler und Computer zwar ihre Figuren, doch die Darstellung auf dem Bildschirm ändert sich so, daß die weißen Figuren unten stehen und die schwarzen oben. `Set Time Limit` ändert während des Spiels die maximale Bedenkzeit des Computers. Das Programm öffnet einen Requester, in den der neue Wert eingetragen wird. `Skip Move` ist nach den Schach-Regeln eigentlich illegal: Der Spieler setzt einen Zug aus. `Undo Move` ist für einen mittelmäßigen Spieler ebenso illegal wie nützlich: Stellt sich nämlich heraus, daß ein Zug einen schweren Fehler darstellte, ist es möglich, diesen Zug rückgängig zu machen, bzw. durch mehrmaliges Aufrufen auch mehrere vorangehende Züge. Mit `Hint` gibt der Computer dem Spieler einen Tip, wenn dieser absolut nicht mehr weiß, wohin er ziehen soll.

Mit dieser gut gelungenen grafischen Umsetzung ist das Programm `Chess` nun endlich ein richtiges Amiga-Programm und ein ernstzunehmender Schachpartner, zumindest für die Amigos, die noch nicht Großmeister-Status erreicht haben. Es ist auf der Fishdisk 205 zu finden.

© 1990 technicSupport

Conman

Editierhilfe für das CLI

AUFRUF

Nach erfolgter Installation der nötigen Dateien wird `Conman` vom CLI aus mit

`Conman` -c|-r [-b⟨*xxxx*⟩] [-l⟨*xxx*⟩] [-o] [-s] [-t] [-w] [-d]

gestartet. Dieser Aufruf sollte in die `startup-sequence` eingebunden werden, damit `Conman` jederzeit zur Verfügung steht. Der Aufruf über Icons ist prinzipiell möglich, in der Praxis aber sehr umständlich. Zum ersten Ausprobieren des Programms können Sie jedoch mit den Icons auf Fishdisk 165 arbeiten. Zuvor müssen Sie allerdings in einem CLI-Fenster die Zuweisung

```
assign L: AmigaLibDisk165:Conman
```

vornehmen, damit `Conman` „seinen" `Conhandler` findet. Öffnen Sie dann die `Conman`-Schublade, starten Sie das `LoadLib`-Programm mit einem Doppelklick auf das entsprechende Icon und verfahren Sie ebenso mit dem `Conman`-Piktogramm. Wenn Sie nun ein *neues* CLI-Fenster öffnen, steht Ihnen `Conman` zur Verfügung.

OPTIONEN

Die folgenden Optionen betreffen die generelle Arbeitsweise von `Conman` und sollten schon beim Programmaufruf übergeben werden. Die Optionen `-o`, `-s`, `-t` und `-w` können auch nachträglich mit Hilfe des Programms `SetCMan` gesetzt bzw. geändert werden. Hierzu ist einzugeben.

`SetCMan` [-o] [-s] [-t] [-w]

`-b`⟨*xxxx*⟩ Die Größe des Eingabepuffers in Byte wird festgesetzt. Ohne diese Option wird die Voreinstellung auf 488 Byte übernommen, die für etwa 20 bis 30 Befehlszeilen ausreichen dürfte.

 © 1990 technicSupport

-l⟨*xxx*⟩ Eine Eingabezeile wird erst dann in den Puffer übernommen, wenn ihre Länge der Zahl ⟨*xxx*⟩ entspricht. Voreingestellt ist ein Wert von 1.

-c `Conman` wird als Ersatz für das `CON:`-Gerät installiert.

-r `Conman` ersetzt das `RAW:`-Gerät.

-o Der Überschreibmodus wird aktiviert.

-s Änderungen des Editiermodus mittels [CTRL]-[A] bleiben über die aktuelle Eingabezeile hinaus gültig.

-t Der *True history*-Modus (s.u.) wird eingeschaltet.

-w Der Eingabepuffer wird in einer Schleife abgearbeitet.

-d Wird ein über `ConMan` geöffnetes Fenster durch Anklicken des [●]-Gadgets geschlossen, sendet `ConMan` an alle Tasks, die noch Daten an den Handler schicken das Abbruchsignal [CTRL]-[C]. Das Senden dieses Signals kann mit der `-d`-Option verhindert werden.

ZWECK

`Conman` ist eine sehr universelle Arbeitshilfe. `Conman` wirkt grundsätzlich auf alle Fenster, die über „CON:“ geöffnet werden, empfiehlt sich aber besonders als Editierhilfe für das CLI, dessen Eingabekomfort bekanntlich zu wünschen übrig läßt. Einige dieser Aufgaben erfüllt auch das `NewCon:`-Gerät der Workbench 1.3, das jedoch keine ernsthafte Konkurrenz zu `ConMan` darstellt.

`ConMan` stellt auf Wunsch einige andere Geräte zur Verfügung, von denen das Pipe-Gerät `Pip:` hervorzuheben ist, weil es die einzige Pipe ist, mit der die ARP-Shell (an anderer Stelle in diesem Buch beschrieben) zusammenarbeitet.

BESCHREIBUNG

Bevor Sie die Vorzüge von `Conman` nutzen können, müssen die erforderlichen Dateien in ihre „angestammten“ Verzeichnisse auf der Workbench kopiert werden. Das Programm selbst gehört in die `c`-Schublade, die Bibliotheksdatei `Conlib.lib` unter Umbenennung in

`Conhandler.library` in das `Libs`- und der Gerätetreiber in das `L`-Verzeichnis:

```
cd AmigaLibDisk165:Conman
copy Conman <Zieldiskette>:c

copy Conlib.lib <Zieldiskette>:Libs/Conhandler.library
copy Conhandler <Zieldiskette>:L
```

⟨*Zieldiskette*⟩ steht hier für den Namen Ihrer Workbench-Diskette. Wenn Sie mehr als ein Laufwerk zur Verfügung haben, können Sie auch die Gerätebezeichnung des Laufwerks, in dem sich die Zieldiskette befindet, angeben (z.B. df0:).

`Conman` wird von einigen Hilfsprogrammen unterstützt, die ebenfalls in diesem Text erwähnt werden, z.B. `SetCMan` oder `History`. Falls Sie einige dieser Programme benötigen, müssen Sie sie analog zu der Programmdatei „Conman" in das `c`-Directory der Workbench kopieren.

Überprüfen Sie nun den Erfolg der Installation, indem Sie aus dem CLI heraus

```
Conman -c
```

eingeben und anschließend ein *neues* CLI-Fenster öffnen, in dem Sie einige der Programmfunktionen ausprobieren können. Das Öffnen eines neuen Fensters ist unbedingt notwendig, da `Conman` nicht rückwirkend auf bereits vorhandene Fenster wirkt.

Editierfunktionen

`Conman` bietet eine ganze Reihe von Editierfunktionen, die in erster Linie den Komfort bei der Korrektur von Eingabefehlern erhöhen. Mit [←] bzw. [→] können Sie sich innerhalb der aktuellen Eingabezeile nach links bzw. rechts bewegen, in Verbindung mit der [Shift] -Taste bewirken die Cursortasten einen Sprung nach links oder rechts zum jeweils nächsten Wortanfang. Mit [Del] und [Backspace] löschen Sie das Zeichen unter dem bzw. links vom Cursor, wie es auch bei Textverarbeitungen üblich ist. Zusätzlich stehen Ihnen weitere Sondertasten zur Verfügung:

 © 1990 technicSupport

[CTRL]-[A]	Umschalten zwischen Überschreib- und Einfügemodus. Sofern der Aufruf von `Conman` nicht mit der `s`-Option erfolgte, wird in der nächsten Zeile wieder die Voreinstellung gültig.
[CTRL]-[#]	Aktiviert in jedem Fall den Einfügemodus.
[CTRL]-[Y]	Löscht alle Zeichen ab der Cursorposition.
[CTRL]-[U]	Löscht alle Zeichen links vom Cursor.
[CTRL]-[X]	Löscht die gesamte Zeile.
[CTRL]-[B]	Löscht den Eingabepuffer der History-Funktion.
[CTRL]-[-]	Macht das versehentliche Aufrufen einer History-Taste rückgängig.
[CTRL]-[S]	Hält Bildschirmausgaben an.
[CTRL]-[Q]	Läßt die Bildschirmausgabe weiterlaufen.
[CTRL]-[W]	Löscht das gesamte Fenster.
[F7]	Löscht links vom Cursor bis zum nächsten Leerzeichen.
[F8]	Löscht rechts vom Cursor bis zum nächsten Leerzeichen.

History-Funktionen

`Conman` speichert sämtliche Tastatureingaben zeilenweise in einem Puffer, aus dem sich diese Eingaben nach Bedarf wieder abrufen lassen. Mit [↑] und [↓] können Sie sich in Ihren „gesammelten" Eingaben rückwärts und vorwärts bewegen. Eine früher getätigte Eingabe oder eine ganze Eingabefolge läßt sich ohne Tipparbeit erneut aufrufen, indem sie mit [↑] zurückgeholt und einfach mit [Return] bestätigt wird. Selbstverständlich können vor dem erneuten Aufruf beliebige Änderungen an dieser Zeile vorgenommen werden.

Mit Hilfe der Funktionstasten [F5] und [F6] kann der Eingabepuffer nach bestimmten Kommandozeilen durchsucht werden. Hierzu muß der Anfang der gesuchten Befehlszeile eingegeben und anschließend

[F5] für Suchen in Vorwärts- bzw. [F6] für Suchen in Rückwärtsrichtung betätigt werden. Ist die Suche erfolgreich, wird die entsprechende Befehlsfolge in die aktuelle Eingabezeile übernommen.

Eine zurückgeholte Eingabe, die unverändert wieder abgerufen wird, wird normalerweise nicht ein weiteres Mal in den Puffer übernommen, wodurch Sie die Möglichkeit haben, auch längere Kommandofolgen wiederholt zu übernehmen. Sie können allerdings auch veranlassen, daß grundsätzlich jede Eingabezeile gespeichert wird, indem Sie Conman mit der Option `-t` (*true history mode*) aufrufen. Die Option `-w` sorgt dafür, daß `Conman` den Puffer zyklisch abarbeitet, was bedeutet, daß nach dem Erreichen der letzten gespeicherten Zeile die nächste Zeile vom anderen Ende des Puffers geholt wird.

Mit dem Hilfsprogramm `History` läßt sich der Inhalt des Eingabepuffers betrachten, abspeichern oder zurücksichern. Zum Betrachten genügt die einfache Eingabe von

```
History
```

während zum Abspeichern ein Umleiten der Ausgabe mittels

`History` > ⟨*Dateiname*⟩

nötig ist. Diese Datei kann mit

`History` ⟨*Dateiname*⟩

wieder in den Puffer zurückgeladen werden, natürlich nur aus einem Fenster heraus, in dem `Conman` aktiviert ist.

Window-Handling

Besonders wenn man mit mehreren Fenstern gleichzeitig arbeitet, ist es oft sehr lästig, zur Maus greifen zu müssen, um diese Fenster in der einen oder anderen Weise zu manipulieren. Einige häufig erforderliche Operationen lassen sich bei `Conman` über Funktionstasten ausführen. Mit [F1] läßt sich das aktuelle Fenster auf Minimalgröße verkleinern bzw. bei erneuter Betätigung auf genau die Größe bringen, die es beim Aufruf hatte; mit [F2] läßt sich zwischen dieser Größe und dem maximal möglichen Format hin- und herschalten.

 © 1990 technicSupport

Dies funktioniert jedoch nur, wenn das entsprechende Fenster aktiviert ist und über ein „Zieh"-Gadget (in der rechten unteren Ecke) verfügt. [F10] schaltet das Fenster abwechselnd in den Vorder- bzw. Hintergrund, [F9] bewirkt dasselbe für den Screen, auf dem sich das `Conman`-Fenster befindet.

Öffnen neuer Fenster

`Conman` akzeptiert die Standardkonventionen für das Öffnen neuer Fenster, z.B. über den Befehl

```
newcli con:10/10/200/100/NeuesFenster
```

Auf diese Weise geöffnete Fenster besitzen eine Reihe von Eigenschaften, die in der folgenden Tabelle zusammengefaßt sind. `Conman` bietet Ihnen nun die Möglichkeit, diese Parameter zu ändern. Hierzu muß dem Titel des Fenster ein weiterer Schrägstrich nachgestellt werden, gefolgt von Optionen, die dafür sorgen, daß die betreffende Voreinstellung durch die entgegengesetzte Eigenschaft ersetzt wird.

Option	Voreinstellung
`a`	Fenster wird automatisch aktiviert
`b`	Fenster erscheint im Vordergrund
`c`	Fenster besitzt kein Schließ-Gadget
`d`	Fenster hat Depht-Gadgets für Vorder-/Hintergrund
`m`	Fenster besitzt Ziehleiste (*Drag-Gadget*)
`n`	Fenster ist mit Rahmen versehen
`r`	*SmartRefresh* (mit `r`-Option dagegen: *SimpleRefresh*)
`s`	Fenster hat *Sizing-Gadget* (zum Vergrößern)
`z`	Fenster ist kein Gimmezerozero

Ein nicht sofort aktiviertes Fenster ohne Rahmen und ohne Ziehleiste läßt sich demnach mit dem Befehl

```
newcli con:10/10/200/100/NeuesFenster/anm
```

öffnen.

PROBLEME

Wird `Conman` zusammen mit anderen Programmen verwendet, die die Funktionstasten anders belegen, so gehen alle Funktionen Conmans, die über diese Tasten aufzurufen sind, verloren. Probleme kann es auch beim Aufruf extrem kleiner Fenster (z.B. `con:0/0/5/10/GanzKlein`) geben, dafür ist jedoch nicht `Conman` verantwortlich, sondern das Amiga-Betriebssystem.

HERKUNFT

Die besprochene Version 1.3 des Programmes sowie die erwähnten Hilfsprogramme findet man auf Fishdisk 165. Das Programm ist Shareware und sollte mit $10 honoriert werden — gewiß keine zu hohe Summe für ein so hervorragendes Programm! Autor ist

```
William S. Hawes
P.O. Box 308
Maynard, MA 01754
(617) 568-8695
```

VERWEISE

`Ash` (Fishdisk 018), `Cled` (Fishdisk 081), `Csh` (Fishdisk 107), `MyCLI` (Fishdisk 004), `NewShell` (Panorama 11a), `Shell` (Fishdisk 014), `SiliCon` (Fishdisk 102), `NewCon` (Workbench 1.3)

© 1990 technicSupport

DFC2

Kopiert und formatiert Disketten

AUFRUF

`DFC2` kann von der Workbench aus angeklickt werden. Der Aufruf aus dem CLI erlaubt die Übergabe mehrerer Optionen.

`DFC2` [f ⟨*Disk*⟩] [t ⟨*Disks*⟩] [v|nov] [b|nob] [n ⟨*Name*⟩]

OPTIONEN

Den oben angegebenen Optionen darf ein Spiegelstrich vorangestellt werden. Die Abkürzungen dürfen auch ausgeschrieben werden. Der vollständige Wortlaut ist der folgenden Auflistung zu entnehmen.

`f[rom]` Dieser Option muß die Angabe ⟨*Disk*⟩ für das Quell-Laufwerk folgen. Die Laufwerksangaben dürfen in der Form `df0:`, `df1:`..., aber auch in der Form `0`, `1`, `2`, `3` erfolgen.

`t[o]` Dieser Option folgen die Bezeichnungen ⟨*Disks*⟩ für die Ziel-Laufwerke.

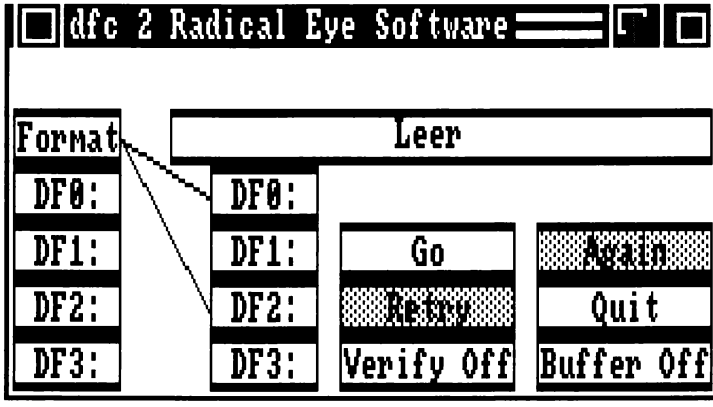
`v[erify]` Das Kopieren erfolgt mit Überprüfung des Schreibvorganges. Grundeinstellung.

`nov[erify]` Schaltet die Überprüfung des Schreibvorganges aus.

`b[uffer]` Es wird versucht, möglichst viele Daten der Quelldiskette in einem Puffer zu behalten. Das kostet Speicherplatz, spart aber Zeit, wenn eine Diskette sehr oft kopiert werden soll, weil sie nicht immer wieder neu eingelesen werden muß. Beim Kopieren mit nur einem Laufwerk muß der Puffer eingeschaltet sein.

`nob[uffers]` Der Puffer wird abgeschaltet. Grundeinstellung.

`n[ame]` Dieser Option folgt der gewünschte Name, den formatierte Disketten erhalten sollen. Zugleich signalisiert diese Option, daß nicht kopiert, sondern formatiert werden soll.



DFC2 formatiert mehrere Disketten gleichzeitig

BESCHREIBUNG

Mit DFC2 lassen sich auf bequeme Weise Disketten formatieren und kopieren, wobei alle Einstellungen per Mausklick auf der Workbench gewählt werden können. Die Einstellungen lassen sich auch über die Optionen vornehmen, wenn DFC2 aus dem CLI aufgerufen wird. DFC2 ist multitaskingfähig, man kann also während des Kopiervorganges mit anderen Programmen weiterarbeiten und sogar auf Laufwerke zugreifen, falls diese für den Kopiervorgang nicht benötigt werden.

Nach dem Aufruf erscheint DFC2 mit einem Fenster, das zahlreiche Gadgets enthält. Durch Anklicken eines der fünf Gadgets am linken Rand wird das Quell-Laufwerk bzw. Formatieren ausgewählt. Rechts daneben werden die Ziel-Laufwerke selektiert, d. h. die Laufwerke, auf die der Inhalt der Quell-Diskette kopiert bzw. die formatiert werden sollen — das bedeutet, daß sich mit DFC2 sogar mehrere Disketten gleichzeitig formatieren lassen. Zwischen den linken und rechten Gadgets erscheinen dann kleine Linien, an denen sich die

 © 1990 technicSupport

vorgenommene Einstellung ablesen läßt. Das Anklicken von Gadgets, zu denen kein Laufwerk existiert, bleibt wirkungslos.

Weitere Einstellungen lassen sich über die Gadgets an der rechten Fensterseite vornehmen, die sich auch über die Tastatur betätigen lassen, wenn das DFC2-Fenster aktiviert ist:

[Verify On] oder [V]	Die Überprüfung des Schreibvorganges wird ein- bzw. ausgeschaltet.
[Buffer Off] oder [B]	Der Puffer wird ein bzw. ausgeschaltet.
[Go] oder [Return]	Startet den Kopier- bzw. Formatiervorgang.
[Again] oder [A]	Wiederholt den Kopiervorgang und berücksichtigt dabei die im Puffer befindlichen Daten.
[Retry] oder [R]	Tritt beim Schreiben oder Lesen einer Spur ein Fehler auf, dann können mit diesem Gadget Lese- und Schreibvorgang wiederholt werden.
[Quit] oder [Q]	Bricht einen laufenden Kopier- bzw. Formatierprozeß ab. Läuft kein solcher Prozeß, wird das Programm beendet.

Oberhalb dieser Felder liegt noch ein längeres Textgadget, in welches der Name eingetragen wird, den zu formatierende Disketten erhalten sollen. Ohne Namensangabe erhalten die Disketten automatisch den Namen Empty.

Wie bei jedem anderen Kopierprogramm gilt auch für DFC2, daß man sich vor dem Start des Kopier- bzw. Formatiervorgangs sorgfältig vergewissern sollte, daß sich die richtigen Disketten in den angewählten Laufwerken befinden. Obwohl sowohl Kopieren als auch Formatieren ohne *Verify* schneller geht, rät der Autor davon ab, auf diese Überprüfung zu verzichten. Schreibfehler aufgrund defekten Diskettenmaterials werden dann nicht mehr erkannt, und es ist wenig sinnvoll, Sicherheitskopien anzulegen, von denen man nicht sicher ist, ob sie überhaupt in Ordnung sind. DFC2 kopiert nur Disketten in Standard-AmigaDOS-Format, es kann also keine mit Kopierschutz ausgestatteten Disketten kopieren.

HERKUNFT

DFC2 findet man auf der TBAG 29. Die Adresse des Autors:

Tomas Rokicki
Box 2081
Stanford, CA 94309

VERWEISE

A-Copier 1.3 (ACS 100), BFormat (RPD 118), FFormat1.1 (TBAG 14), PCopy (Fishdisk 151), TurboBackUp (Fishdisk 139), QuickCopy 1.01 (ACS 79), QuickCopy 1.0 EA (Franz 002)

© 1990 technicSupport

DiskMan

Konfigurierbarer Disk-Manager

AUFRUF

`DiskMan` läßt sich durch Anklicken des Icons auf der Workbench oder durch Eingabe des Programmnamens im CLI starten.

ZWECK

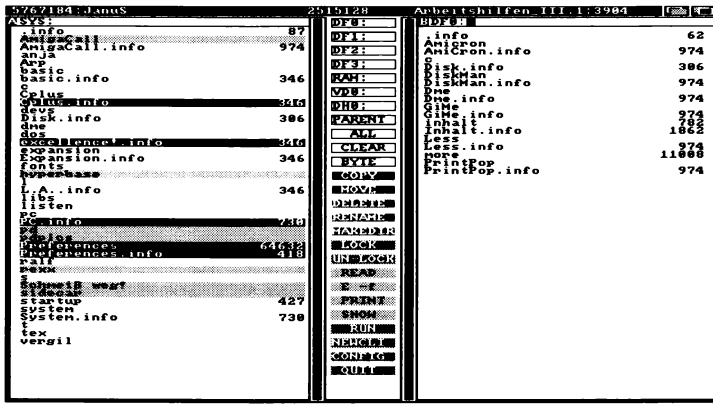
`DiskMan` ermöglicht es, alle wichtigen CLI-Befehle, die sich auf die Arbeit mit Disketten beziehen, komfortabel mit der Maus auszuführen. Dabei müssen sich die Operationen keineswegs nur auf die Diskettenlaufwerke beschränken. Ansprechbar sind selbstverständlich auch alle Arten von Festplatten und RAM-Disks, egal ob diese `DH0:`, `JH0:`, `FAST:`, `RAM:`, `VD0:` oder `CARD:` etc. heißen.

BESCHREIBUNG

Vielen sind wahrscheinlich die schon etwas betagten kommerziellen Programme `CLImate` oder `Zing!` bekannt. Den Vergleich mit diesen, gemessen an der Leistung doch recht teuren Tools braucht `DiskMan` nicht zu scheuen. Funktionalität, Übersicht, Bedienung und Flexibilität sprechen eindeutig für ihn. Negativ ist lediglich zu bemerken, daß `DiskMan` nur den amerikanischen NTSC-Bildschirm ausnutzt — und das auch noch im flimmernden *Interlace*-Modus. Dieses läßt sich, genau wie die Wahl der Farben, auch so ohne weiteres nicht verstellen. Hierzu müßte das Programm geändert werden, was aber durch das Fehlen des Sourcecodes nicht ganz einfach ist.

Die Arbeitsweise von DiskMan

`DiskMan` begrüßt Sie mit einer vertikalen Befehlsleiste in der Mitte und zwei Verzeichnis-Anzeigefenstern, in deren Kopfzeilen auf der linken Seite ein `A` und auf der rechten Seite ein `B` zu sehen sind. Einer dieser beiden Buchstaben ist immer heller dargestellt als der andere, woran Sie das *aktuelle* Verzeichnis erkennen. Durch Klicken



DiskMan: Viele CLI-Kommandos per Mausklick

in das jeweils nicht aktive Anzeigefenster bzw. dessen Scrollbalken können Sie diesen Zustand leicht ändern.

Es kann immer nur *ein* Verzeichnis das *zur Zeit aktuelle* sein! Alle Operationen verhalten sich relativ zu diesem Status. Ein [DELETE] kann also nur etwas im aktuellen Verzeichnis löschen und ein [COPY] kopiert, getreu der Schablone „FROM ... TO ...“, immer vom aktuellen zum auf der gegenüberliegenden Seite dargestellten Verzeichnis. Während einer Operation können Sie `DiskMan` durch einfaches Klicken auf den Bildschirm unterbrechen.

Anzeigen von Verzeichnissen

Klicken Sie einfach auf einen der Verzeichnis-Buttons oben in der Befehlsleiste, und `DiskMan` zeigt Ihnen das Basisverzeichnis einer beispielsweise im Laufwerk `DF0:` befindlichen Diskette an. Hierbei werden die Verzeichnisse rot und die Dateien weiß angezeigt. Ist eine Datei geschützt, so wird diese grün dargestellt. Außerdem

 © 1990 technicSupport

wird jeweils die Dateigröße in Byte angegeben. In der Titelzeile des Bildschirmes, in der auch der aktuelle Arbeitsstatus und eventuelle Fehlermeldungen erscheinen, steht über beiden Anzeigefeldern der Name der jeweiligen Diskette sowie die noch verfügbaren Bytes. In der Mitte dieser Zeile finden Sie auch noch den freien Speicher Ihres Systems.

Möchten Sie sich nun den Inhalt eines Verzeichnisses Ihrer Diskette anschauen, so können Sie dieses mit einem Doppelklick auf das entsprechende Verzeichnis tun. So verschwindet das vorher angezeigte Basisverzeichnis zugunsten des soeben geöffneten Unterverzeichnisses. Um nun wieder einen Schritt in der Verzeichnisstruktur Ihrer Diskette zurückzugehen, klicken Sie einfach auf [PARENT] in der Befehlsleiste, was dem CLI-Befehl „`cd /`“ entspricht, und schon sind Sie wieder im Basisverzeichnis.

Eine andere Variante, sich den Inhalt eines Verzeichnisses anzusehen, ist es, zuerst das zu öffnende Verzeichnis anzuklicken und dann möglichst zügig in das gegenüberliegende Anzeigefeld zuklicken. Hier wird dann das geöffnete Verzeichnis angezeigt. So können Sie z.B. direkt nacheinander alle Unterverzeichnisse, die im Basisverzeichnis stehen, ansehen, ohne jedesmal wieder mit [PARENT] zurückzugehen.

Die dritte Möglichkeit ist zwar keine reine Mausoperation, dafür aber sehr effektiv. Wie Sie vielleicht schon bemerkt haben, erscheint jeweils in der Kopfzeile der Anzeigefenster der komplette Pfadname als Klartext. Hier können Sie, falls der Cursor noch nicht sichtbar ist, hineinklicken und den Pfad beliebig verändern. Mit [Amiga]-[X] löschen Sie den gesamten Inhalt der Zeile. Haben Sie einen Pfadnamen geändert oder neu eingegeben, so müssen Sie diesen mit [Return] bestätigen. Der Vorteil dieser Methode besteht u.a. darin, daß Sie bei bekanntem Pfad direkt in beliebig tiefe Unterverzeichnisse gelangen können, ohne vorher die darüberliegenden Verzeichnisse laden zu müssen. Außerdem können Sie so auch neue Geräte ansprechen, für die Sie noch keine Verzeichnis-Buttons definiert haben.

Ein Tip: `DiskMan` arbeitet hervorragend mit dem Floppybeschleuniger `FaccII` von der Firma ASDG Inc. zusammen. Hierdurch bleibt

ein einmal eingelesenes Inhaltsverzeichnis einer Diskette im Speicher und muß nicht jedesmal wieder neu geladen werden. Das spart viel Zeit!

Markieren von Dateien und Verzeichnissen

Bevor Sie einen Befehl ausführen können, müssen Sie eine oder mehrere Dateien bzw. Verzeichnisse auswählen. Dies geschieht durch einfaches Anklicken in der Liste, wodurch Sie auch gleichzeitig das entsprechende Verzeichnis als aktuelles erklären. Markierte Dateien werden hellblau und markierte Verzeichnisse violett unterlegt. Löschen können Sie diese Markierungen durch erneutes Anklicken. Mit den beiden Buttons [ALL] und [CLEAR] sind Sie in der Lage alle Dateien und Verzeichnisse der aktuellen Seite auf einmal zu markieren bzw. die Markierungen zu löschen.

Die Funktionen der Befehlsleiste

[DF0:] Zeigt den Inhalt von `DF0:` im aktuellen Anzeigefenster an.

[DF1:] Zeigt den Inhalt von `DF1:` im aktuellen Anzeigefenster an.

[DF2:] Zeigt den Inhalt von `DF2:` im aktuellen Anzeigefenster an.

[DF3:] Kann mit [CONFIG] geändert werden. Zeigt den Inhalt von `DF3:` im aktuellen Anzeigefenster an.

[RAM:] Zeigt den Inhalt von `RAM:` im aktuellen Anzeigefenster an.

[VD0:] Kann mit [CONFIG] geändert werden. Zeigt den Inhalt von `VD0:` im aktuellen Anzeigefenster an.

[DH0:] Kann mit [CONFIG] geändert werden. Zeigt den Inhalt von `DH0:` im aktuellen Anzeigefenster an.

© 1990 technicSupport

[PARENT] Geht in der Verzeichnisstruktur einen Schritt zurück, entspricht also dem CLI-Befehl `cd /`.

[ALL] Markiert alle Dateien und Verzeichnisse im aktuellen Anzeigefenster.

[CLEAR] Löscht alle Markierungen im aktuellen Verzeichnis.

[BYTE] Addiert alle Größen der markierten Dateien im aktuellen Anzeigefenster und zeigt die Summe in der Titelzeile an.

[COPY] Kopiert alle im aktuellen Anzeigefenster markierten Verzeichnisse und Dateien in das Verzeichnis der gegenüberliegenden, nicht aktuellen Seite. Die Verzeichnisse werden dabei vollständig, d.h. mit komplettem Inhalt, kopiert. Ist in `A` und `B` dasselbe Verzeichnis enthalten, erstellt `DiskMan` ein Duplikat mit einem „`.BAK`" als Endung.

[MOVE] Bewegt alle im aktuellen Anzeigefenster markierten Verzeichnisse und Dateien in das Verzeichnis der gegenüberliegenden, nicht aktuellen Seite. Die Verzeichnisse werden dabei vollständig, d.h. mit komplettem Inhalt, bewegt. Befinden sich die Verzeichnisse in `A` und `B` beide auf einer Diskette, findet nur ein **`Rename`** statt, d.h. die entsprechenden Pfadnamen werden geändert. Befinden sie sich auf unterschiedlichen Disketten, wird zuerst kopiert und dann im aktuellen Verzeichnis gelöscht.

[DELETE] Löscht alle im aktuellen Anzeigefenster markierten Verzeichnisse und Dateien. Die Verzeichnisse werden dabei vollständig, d.h. mit komplettem Inhalt, gelöscht.

[RENAME] Benennt alle im aktuellen Anzeigefenster markierten Verzeichnisse und Dateien um. Die neuen Namen werden dabei über einen Requester abgefragt. Wird als Name ein kompletter Pfad angegeben, so verhält sich dieser Befehl wie `Move`.

[MAKEDIR] Erstellt im aktuellen Anzeigefenster ein neues Verzeichnis. Der Name wird durch einen Requester abgefragt. Es darf auch ein kompletter Pfad eingegeben werden.

[LOCK] Schützt alle im aktuellen Anzeigefenster markierten Dateien gegen Löschen bzw. Überschreiben. Setzt die Flags auf `R-E-` und entspricht dem CLI-Befehl `Protect`.

[UN-LOCK] Entsichert alle im aktuellen Anzeigefenster markierten Dateien. Setzt die Flags auf `RWED`.

[READ] Zeigt alle im aktuellen Anzeigefenster markierten Dateien im `ASCII`- bzw. `HEX`-Format an. Das Anzeigeformat wählt `DiskMan` automatisch.

[EDIT] Kann mit [CONFIG] geändert werden. Mit diesem Button können Sie Ihren Editor aufrufen und alle markierten Dateien editieren. z.B. Ed, Edit, Emacs, TxEd, CED, NewZAP, etc.

[PRINT] Kann mit [CONFIG] geändert werden. Hiermit können Sie z.B. ein beliebiges `Print`-Utility aufrufen und alle markierten Dateien damit ausdrucken. Geeignete Programme sind ausführlich in Band I dieser Reihe beschrieben. Sie finden diese auf den Disketten zum Buch I oder auf verschiedenen Public Domain Disketten z.B. Fishdisk 32 oder RPD 42. Selbstverständlich können Sie auch andere Programme mit diesem Button starten — z. B. den CLI-Befehl `Execute`.

[SHOW] Kann mit [CONFIG] geändert werden. Mit diesem Button können Sie sich z.B. alle markierten Bilddateien ansehen. Vorausgesetzt, Sie haben ein `Show`-Utility auf Ihrer Diskette. Solche Programme finden Sie auf verschiedenen Public Domain Disketten z. B. auf RPD 56.

[RUN] Startet alle im aktuellen Anzeigefenster markierten Dateien, wobei etwaige Argumente in einem Requester eingegeben werden können.

© 1990 technicSupport

[NEWCLI] Kann mit [CONFIG] geändert werden. Öffnet ein neues CLI-Fenster oder startet bei entsprechender Belegung ein beliebiges Programm, z.B. ein Kopierprogramm.

[QUIT] Beendet `DiskMan`.

[CONFIG] Mit diesem Button lassen sich alle in folgender Tabelle dargestellten Befehle neu definieren und abspeichern, so daß diese beim nächsten Start von `DiskMan` automatisch zur Verfügung stehen. Es wird dabei eine Datei mit dem Namen `DM.Configure` im `devs`-Verzeichnis Ihrer Startdiskette angelegt. Ist ein Button mit mehreren Befehlen belegt, so kann innerhalb des Programms durch Anklicken des Buttons mit der rechten Maustaste zwischen den einzelnen Belegungen hin- und hergeschaltet werden. Hier ein Beispiel für die Belegung einiger Buttons:

[DF3:] — DF0:c, DF1:fonts, SYS:, etc.

[VD0:] — CARD:, c:, devs:, fonts:, libs:, etc.

[DH0:] — DH1:, DH2:, FAST:, JH0:, etc.

[EDIT] — CED, Ed, Emacs, TxEd, NewZAP, etc.

[PRINT] — Print, Execute, FixHunk, etc.

[SHOW] — Show, Showprint, View, Play, etc.

[NEWCLI] — NewShell, NewCLI, XCopy, etc.

Arc v — Arc v, Arc l, LArc (für Archiv-Menü)

Arc x — Arc x, UnArc (für Archiv-Menü)

Das Archiv-Menü

`DiskMan` verfügt neben den Standard-CLI-Funktionen auch über die Möglichkeit, die sehr nützlichen Programme `Arc` und `Zoo` mit Hilfe der Maus zu benutzen. Das Pull-Down-Menü befindet sich rechts in der Titelzeile. Beide Programme dienen zur Reduzierung des Speicherbedarfs beliebiger Dateien. Diese Methode findet besonders häufig bei der Daten-Fern-Übertragung (DFÜ) ihre Anwendung und dient dazu, Telefongebühren und Übertragungszeiten zu

sparen. Beide Programme sind Public Domain und bereits ausführlich in den beiden ersten Bänden dieser Buchreihe beschrieben worden. Sie finden eine genaue Beschreibung zu `Arc` im ersten und zu `Zoo` im zweiten Band. Bevor Sie diese wirklich sehr leistungsfähigen Utilities aber benutzen können, müssen Sie sie erst auf Ihre Arbeitsdiskette kopieren.

`Archive` Archiviert alle im aktuellen Anzeigefenster markierten Dateien, wobei Sie Pfad und Namen in einem Requester eingeben können. Die Endung „`.arc`“ wird automatisch an den Dateinamen angehängt, damit Sie immer wissen, mit welchem Programm Sie dieses Archiv erstellt haben.

`Arc Extract` Extrahiert alle markierten `.arc`-Archive im aktuellen Anzeigefenster. Den Ausgabepfad können Sie in einem Requester eingeben. Den Befehl, den `DiskMan` bei Anwahl dieses Menüpunktes ausführen soll, können Sie mit [CONFIG] definieren. Meistens wird dies `Arc x` sein.

`Arc List` Zeigt den Inhalt aller markierten `.arc`-Archive an. Den Befehl, den `DiskMan` bei Anwahl dieses Menüpunktes ausführen soll, können Sie mit [CONFIG] definieren. Meistens wird dieses sicher `Arc l` sein.

`Zoo-ive` Archiviert alle im aktuellen Anzeigefenster markierten Dateien, wobei Sie Pfad und Namen in einem Requester eingeben können. Die Endung „`.zoo`“ wird automatisch an den Dateinamen angehängt, damit Sie immer wissen, mit welchem Programm Sie dieses Archiv erstellt haben.

`Zoo Extract` Extrahiert alle im aktuellen Anzeigefenster markierten `.zoo`-Archive. Den Ausgabepfad können Sie in einem Requester eingeben.

`Zoo List` Zeigt den Inhalt aller im aktuellen Anzeigefenster markierten `.zoo`-Archive an.

 © 1990 technicSupport

HERKUNFT

`DiskMan` 2.0 stammt von der RPD 64. Es ist ein Shareware-Programm. Sollte Ihnen dieses Programm also gut gefallen und möchten Sie damit arbeiten, so schicken Sie einfach $10 an den Autor:

`Greg Cunningham`
`2014 Encino White`
`San Antonio, TX 78259, USA`

VERWEISE

Auf ähnliche Weise arbeiten die inzwischen nicht mehr zählbaren Varianten des Programmes `Dir Util`. Eine vergleichbare Benutzerführung weist auch das Progamm `SmartDisk` (Panorama 028d) auf, das zusätzlich einen einfachen Diskettenmonitor beinhaltet.

Dme

Ein Editor zum Maßschneidern

AUFRUF

`Dme` [-f⟨*Script*⟩] [-b] [-t⟨*N*⟩] [-l⟨*N*⟩] [-w⟨*N*⟩] [-h⟨*N*⟩] {⟨*Dateiname*⟩}

`Dme` kann auch über sein Icon von der Workbench aus aufgerufen werden. In diesem Fall versieht `Dme` jede abgespeicherte Datei ebenfalls mit einem Icon.

OPTIONEN

`-f`⟨*Dateiname*⟩ Ohne diese Option liest `Dme` alle Angaben zur Tastatur- und Menübelegung aus einer Datei namens „S:.edrc“. Mit dieser Option kann eine andere Datei als Script spezifiziert werden.

`-t`⟨*N*⟩ Der Editor öffnet sein Fenster ab Zeile N. Mit Zeile ist hier die Rasterzeile (0–255) gemeint.

`-l`⟨*N*⟩ Das Fenster wird um N Rasterspalten nach rechts verschoben.

`-w`⟨*N*⟩ Die Breite des Fensters beträgt N Pixel.

`-h`⟨*N*⟩ Die Fensterhöhe wird auf N Pixel festgelegt.

Bei der Angabe der Fensterdimensionen muß man sehr vorsichtig sein, da Fehleingaben nicht abgefangen werden und das Betriebssystem in diesen Fällen mit einem Guru kontert.

ZWECK

`Dme` ist ein Texteditor, der neben den reinen ASCII-Zeichen auch andere Codes verarbeiten kann. Tastaturbelegung und Menüfunktionen können vom Anwender völlig frei bestimmt werden. Als besonderes Bonbon stellt `Dme` eine Schnittstelle zu der Interpretersprache `ARexx` bereit.

© 1990 technicSupport

Der Dme-Editor läßt sich nach eigenen Wünschen konfigurieren

BESCHREIBUNG

Leser des zweiten Bandes unserer Reihe werden sich vielleicht fragen, warum nach dem `Med` schon wieder ein Editor besprochen wird. Die Antwort fällt nicht schwer: Ein Editor ist eines der wichtigsten Werkzeuge beim Programmieren und beim Arbeiten auf der Ebene des Betriebssystems. Der Satz dieses Buches erfordert es beispielsweise, alle Rohtexte mehrfach durch einen Editor zu „schieben" und sie mit diesem zu bearbeiten. Für solche Aufgaben kann ein Editor nicht gut genug sein, der `Dme` ist durch seine hohe Geschwindigkeit und Anpassungsfähigkeit für solche Aufgaben gut geeignet.

Ehrlicherweise muß ich zugeben, daß ich mittlerweile auch den `Dme` beiseite gelegt habe und seit einigen Wochen mit dem `CygnusEd Professional` arbeite, einem kommerziellen Produkt, das keinerlei Wünsche mehr offenläßt. Die endgültige Version ist noch weit besser als die Demofassung auf der RPD 040, die noch nicht über den vollen Befehlsumfang verfügt, noch keine deutschen Sonderzeichen akzeptiert und nur Dateien bis 5 KByte abspeichern kann. Den-

noch kann ich jedem nur empfehlen, sich einmal mit dieser Demo auseinanderzusetzen.

Nach diesem kleinen Exkurs — der **Ced** ist doch wirklich so gut, daß ich mich verleiten lasse, hier kostenlose Werbung für ASDG zu machen — nun aber zurück zum **Dme**, der zwar nicht ganz so perfekt ist, dafür aber etwa DM 160.- billiger. Nach einem ersten Aufruf mit

Dme ⟨*Dateiname*⟩

steht man da wie der Ochs vor dem berühmten Scheunentor. Man kann sich zwar mit den Cursortasten und der Maus durch den Text bewegen, aber mehr ist nicht drin. Versucht man, mit der rechten Maustaste ein Menü zu finden, schrumpft das Fenster plötzlich auf die Größe eines Icons. Zufällig stellt man noch fest, daß man mit [Esc] in eine Kommandozeile gelangt, wie man es vom **Ed** her gewohnt ist. Und wie beim **Ed** muß man sich auch hier mit einer stattlichen Zahl von Befehlen auseinandersetzen, bis man den Editor im Griff hat.

Nähere Bekanntschaft mit diesen Befehlen läßt sich am besten dadurch schließen, daß man sich mit der [Esc] -Taste in die Kommandozeile begibt, dort einen der Befehle eingibt und seine Wirkung beobachtet. Fehler- und Statusmeldungen erscheinen in der Titelleiste des Fensters.

Dme ist ein sehr leistungsfähiger Editor und verfügt über sehr viele Befehle, die man jedoch nicht auswendig lernen muß. Vielmehr empfiehlt es sich, die Befehle herauszugreifen, die man wirklich braucht, und diese über die Menüleiste bzw. über Tastenkombinationen verfügbar zu machen. Auf diese Weise läßt sich der Editor ganz nach den eigenen Bedürfnissen und Vorstellungen einrichten. Die Schlüsselbefehle dazu lauten **menuadd**, **map** und **Escimm**. Auch sie lassen sich in der Kommandozeile ausprobieren.

Da es natürlich wenig Sinn macht, nach jedem Laden des Editors erst einmal eine Stunde lang die Arbeitsumgebung einzurichten, lassen sich alle Befehle in einer Scriptdatei ablegen, die im Normalfall unter dem Namen **.edrc** im **S:**-Verzeichnis abgelegt wird. Diese Befehle werden dann beim Aufruf des Editors automatisch ausgeführt.

© 1990 technicSupport

Für die konkrete Gestaltung der Menüs und Tastaturbelegungen läßt sich kein Patentrezept angeben. Man sollte sich aber um eine einheitliche Linie bemühen und sich beispielsweise entweder für deutsche oder für englische Befehle entscheiden. Häufig benutzte Befehle sollten auch über die Tastatur zugänglich sein.

Einrichten einer deutschen Benutzerführung

Das folgende Beispiel einer Scriptdatei demonstriert die Einrichtung einiger wichtiger Grundbefehle. Zur besseren Kommentierung stellen wir diese Datei in einzelnen Teilen vor. Die Einzelbefehle werden später in einer Befehlsübersicht erläutert.

```
menuoff
menuadd Datei (Laden            l) arpload
menuadd Datei (Einfügen         L) arpinsfile
menuadd Datei (Neues Fenster    f) newwindow
menuadd Datei (Speichern        s) saveold
menuadd Datei (Speichern als...S) arpsave
menuadd Datei (Sp. & Verlassen V) (saveold quit)
menuadd Datei (Verlassen        v) quit
map A-l  arpload
map As-l arpinsfile
map A-f  newwindow
map A-s  saveold
map As-s arpsave
map As-v (saveold quit)
map A-v  quit
```

Der Befehl `menuoff` schaltet das Menü ab, was eine schnellere Einrichtung der Menüleiste bewirkt. Mit dem `menuadd`-Befehl richten wir das erste Menü ein, dem wir den Namen „Datei“ geben. Argumente, die Leerzeichen enthalten, müssen in Klammern eingeschlossen werden. Die Befehle `arpload`, `arpsave` und `arpinsfile` stellen Requester zur bequemen Auswahl von Dateien zur Verfügung. Damit sie funktionieren, muß die „Arp.library“ in das `libs:`-Verzeichnis kopiert werden. Diese Library befindet sich unter anderem auf der RPD 122 und der Panorama 27c.

Alle sieben Positionen des ersten Menüs sollen auch über die Tastatur erreichbar sein. Als Schlüsseltaste wollen wir versuchen, konsequent die Amiga-Taste (A) zu benutzen. Der `map`-Befehl richtet uns die Tastaturbelegung ein. Für das Laden einer neuen Datei steht fortan die Kombination [Amiga]-[L] zur Verfügung. Um eine Datei in die bereits bestehende einzufügen, soll zusätzlich die [Shift]-Taste gedrückt werden (As). Reihenfolge: [Amiga], [Shift], [L]. Da wir immer die [Amiga]-Taste verwenden, genügt in den Menüs ein Hinweis auf die Buchstabentaste. So steht im Menü „Laden" ein kleines „l", bei Einfügen ein großes „L". Mit einer solchen Gedächtnisstütze in der Menüleiste lassen sich die Tastaturkombinationen leichter erlernen, die Menüs nehmen dadurch aber auch mehr Platz in Anspruch.

```
menuadd Design (Wordwrap ein) (wordwrap on)
menuadd Design (Wordwrap aus) (wordwrap off)
menuadd Design (Rechter Rand) (Escimm (margin ))
menuadd Design (Einfügen ein) (insertmode on)
menuadd Design (Einfügen aus) (insertmode off)
menuadd Design (Fensterhöhe ) (Escimm (height ))
menuadd Design (Fensterpos. ) (Escimm (topedge ))
```

Das zweite Menü trägt den Namen „Design" und dient zum Ändern einiger Voreinstellungen. Da diese Befehle nicht allzu häufig gebraucht werden, wurde auf die Tastaturäquivalente verzichtet. Der neue Befehl `Escimm` ist von entscheidender Bedeutung für Kommandos, die die Angabe eines Parameters erfordern. Im Menüpunkt „Rechter Rand" sorgt er dafür, daß der Editor in die Kommandozeile springt und diese mit dem Kommando „margin" vorbesetzt. Der Benutzer muß jetzt nur noch die Zahl für den rechten Rand eingeben und [Return] betätigen. Wichtig ist das Leerzeichen hinter „Escimm", ohne dieses kann `Dme` dieses Kommando nicht auswerten. Das Leerzeichen hinter „margin" dient nur dem Komfort: Der Anwender braucht dieses Leerzeichen nicht mehr selbst einzugeben, sondern kann in der Kommandozeile sofort mit der Eingabe des Parameters fortfahren.

© 1990 technicSupport

```
menuadd Suchen (Suchen...   s) (Escimm (find ))
menuadd Suchen (weiter      w) next
menuadd Suchen (zurück      z) prev
menuadd Suchen (Ersetzen... S) (Escimm (findr ))
menuadd Suchen (weiter      W) (Escimm (nextr))
menuadd Suchen (zurück      Z) (Escimm (prevr))
menuadd Suchen (abc=ABC      ) (ignorecase on)
menuadd Suchen (abc!ABC      ) (ignorecase off)
map A-s (Escimm (find ))
map A-w next
map A-z prev
map As-s (Escimm (findr ))
map As-w nextr
map As-z prevr
```

Im dritten Menü finden wir die Operationen zum Suchen und Ersetzen von Textstücken. Die Tastaturäquivalente sind so gehalten, daß für beide Funktionen dieselben Tasten zuständig sind, beim Ersetzen muß jedoch jeweils die [Shift] -Taste gedrückt werden.

```
menuadd Block (Anfang    a) (unblock block)
menuadd Block (Ende      e) block
menuadd Block (Löschen   x) bdelete
menuadd Block (Kopieren  k) bcopy
menuadd Block (Bewegen   b) bmove
menuadd Block (Speichern..) (Escimm (bsave ))
menuadd Block (-> Anfang A) (goto block)
menuadd Block (-> Ende   E) (goto end)
map A-a (pushmark block)
map A-e block
map A-x bdelete
map A-k bcopy
map A-b bmove
map As-a (goto block)
map As-e (goto end)
```

Im Menü für die Blockfunktionen tritt die erste Inkonsequenz auf. Der Buchstabe „L“ ist bereits vergeben, kann somit nicht mehr für das **L**öschen eines Blocks benutzt werden. Man könnte auf die Kom-

bination [CTRL]-[L] zurückgreifen, stattdessen habe ich aber den freien Buchstaben „X“ gewählt. Daß diese Kombination aus dem Rahmen fällt, hat auch seinen Vorteil, denn nun kann diese doch sehr gefährliche Funktion nur angewählt werden, wenn man sich seiner Absichten völlig bewußt ist.

```
menuadd Spezial (Gehe zu Zeile. g) (Escimm (goto ))
menuadd Spezial (Formatieren    F) reformat
menuadd Spezial (ASCII-Zeichen  c) (Escimm (tlate ))
menuadd Spezial (LF->CRLF) (top repeat -1 (last tlate 13 down))
menuadd Spezial (CRLF->LF) (top repeat -1 (last left if c=13
                                                    del down))
```

Im letzten Menü finden sich einige spezielle Funktionen, z.B. für die Eingabe von Steuerzeichen, die nicht über die Tastatur zugänglich sind. Alle diese Sonderzeichen werden von `Dme` als Rechteck dargestellt. Nach ihrer Eingabe läßt sich daher nicht mehr erkennen, welchen Wert dieses Zeichen hat. Mit den beiden letzten Funktionen lassen sich die Zeilenende-Markierungen in der gesamten Datei in einfache *Linefeeds* (LF) oder in Kombinationen aus *Linefeeds* und *Carriage Returns* umwandeln. Dies kann nützlich sein, wenn ein anderer Editor oder eine Textverarbeitung das Zeilenende in einem mit `Dme` erstellten Text nicht erkennen kann.

```
map s-right last
map s-left first
map s-up screentop
map s-down screenbottom
map c-up top
map c-down bottom
map a-up pageup
map a-down pagedown
map A-i insline
map A-del deline
```

Für größere Sprünge im Text werden die Cursortasten in Kombination mit [Shift], [Ctrl] und [Alt] mit einigen Kommandos belegt. Sie sind nur dann effektiv, wenn sie über die Tastatur aufgerufen werden. Das gilt auch für die beiden letzten Kommandos zum Einfügen bzw. Löschen einer Zeile.

© 1990 technicSupport

```
map help (newwindow newfile s:help.dme)
map f2 ((F2 ist mit diesem Text belegt))
wordwrap on

margin 40
height 100
topedge 156
menuon
```

Die [Help] -Taste wird so belegt daß sie ein neues Fenster öffnet und eine Datei namens „help.dme“ aus dem `S:`-Verzeichnis lädt. Diese Datei muß erst noch erstellt werden, sie sollte eine alphabetische Auflistung aller `Dme`-Befehle enthalten. Auf diese Weise hat man immer eine Online-Hilfsfunktion zur Verfügung. Die folgende Zeile soll die Einrichtung von Floskeltasten demonstrieren. Dann wird die Wordwrap-Funktion aktiviert, Größe und Position für neue Fenster werden festgelegt. Zum Schluß wird das gesamte Menü aktiviert.

Als erster Einstieg läßt sich dieses Beispiel mit dem `Ed` erfassen und unter dem Namen „.edrc“ im `S:`-Verzeichnis anlegen. Schon mit diesen wenigen Funktionen läßt sich recht ordentlich arbeiten. Die Datei läßt sich auch leicht nach eigenen Wünschen abändern oder erweitern. Zu diesem Zweck folgt hier eine Übersicht über fast alle Befehle, die `Dme` kennt. Die vollständige Dokumentation findet sich in der Datei „docs/Dme.doc“ auf der Fishdisk.

Die Grundbefehle des Editors

`ARPINSFILE` Fügt Datei an der Cursorposition ein. Die einzufügende Datei wird in einem Requester ausgewählt. Dazu muß die „Arp.library“ installiert sein.

`ARPLOAD` Laden einer Datei über Auswahl-Requester. Die „Arp.library“ muß installiert sein.

`ARPSAVE` Speichert die Datei. Der Name wird in einem Requester gewählt. Wieder muß die „Arp.library“ installiert sein.

`BACK` Cursor um eine Stelle zurück.

`BACKTAB` Cursor um einen Tabulator zurück.

`BCOPY` Fügt markierten Block über der aktuellen Zeile ein, ohne den markierten Block zu löschen.

`BDELETE` Löscht den markierten Block.

`BLOCK` Markiert Anfang oder Ende eines Blockes.

`BMOVE` Löscht markierten Block und fügt ihn über der aktuellen Zeile ein.

`BOTTOM` Sprung an das Textende.

`BS` Cursor um eine Stelle zurück.

`BSAVE` ⟨*Name*⟩ Speichert den markierten Block unter dem angegebenen Namen.

`BSOURCE` Betrachtet den markierten Block als Scriptdatei und führt die Befehle aus. Gut geeignet zum Austesten von Befehlssequenzen.

`CHFILENAME` ⟨*Name*⟩
Ändert den Namen der Datei, mit der gearbeitet wird.

`DEL` Löscht Zeichen unter Cursor.

`DELINE` Löscht die aktuelle Zeile.

`DOWN` Cursor um eine Position nach unten. Ist am Dateiende ohne Wirkung.

`DOWNADD` Wie `DOWN`, am Ende der Datei wird allerdings eine neue Zeile angefügt.

`ESC` Springt in die Kommandozeile und zurück.

`ESCIMM` ⟨*Text*⟩ Springt in die Kommandozeile und besetzt diese mit einem Text vor.

`EXECUTE` ⟨*DOS-Befehl*⟩
Führt CLI-Kommando aus.

`FIND` ⟨*Text*⟩ Sucht nach vorgegebenem Text.

© 1990 technicSupport

`FINDR` ⟨*Text1*⟩ ⟨*Text2*⟩
Sucht nach Text1 und ersetzt ihn durch Text2.

`FINDSTR` ⟨*Text*⟩ Bestimmt den neuen Suchtext.

`FIRST` Sprung zum Anfang der Zeile.

`FIRSTNB` Springt zum ersten Nicht-Leerzeichen einer Zeile.

`GOTO BLOCK` Springt an den Anfang eines markierten Blocks.

`GOTO START` Wie `GOTO BLOCK`.

`GOTO END` Springt an das Ende eines Blocks.

`GOTO` [+/-]⟨*N*⟩ Springt in Zeile N oder um N Zeilen vor- bzw. rückwärts.

`HEIGHT` ⟨*N*⟩ Bestimmt die Höhe eines neuen Fensters in Pixeln (!!).

`ICONIFY` Läßt das Textfenster auf Icongröße schrumpfen. Um es wieder auf die Originalgröße zu bringen, muß das „Icon“ mit der rechten Maustaste angeklickt werden.

`IGNORECASE` on|off|toggle
Bestimmt, ob ein Unterschied zwischen Groß- und Kleinschreibung gemacht wird. *toggle* steht für Umschalten.

`INSERTMODE` on|off|toggle
Umschalten zwischen Einfügen und Überschreiben.

`INSFILE` ⟨*Name*⟩
Fügt die angegebene Datei an der Cursorposition in den bestehenden Text ein.

`INSLINE` Fügt eine Leerzeile ein.

`JOIN` Fügt die folgende Zeile bei der Cursorposition an die aktuelle Zeile an.

`LAST` Sprung zum Zeilenende.

`LEFT` Cursor um eine Position nach links.

© 1990 technicSupport

`LEFTEDGE` ⟨*N*⟩ Setzt linken Rand für ein neues Fenster (in Pixeln).

`MAP` ⟨*Tastenkombination*⟩ ⟨*Befehl*⟩
Legt eine Befehlssequenz auf eine Tastenkombination.

`MARGIN` ⟨*N*⟩ Setzt den rechten Rand auf Spalte N.

`MENUADD` ⟨*Menü*⟩ ⟨*Menüpunkt*⟩ ⟨*Kommando*⟩
Ordnet ein Kommando einem Menüpunkt zu.

`MENUDEL` ⟨*Menü*⟩ ⟨*Menüpunkt*⟩
Entfernt einen Menüpunkt.

`MENUDELHDR` ⟨*Menü*⟩
Entfernt ein komplettes Menü.

`MENUCLEAR` Löscht die ganze Menüleiste.

`MENUON` Aktiviert die Menüleiste.

`MENUOFF` Schaltet die Menüleiste ab.

`NEWFILE` ⟨*Name*⟩
Lädt eine neue Datei.

`NEWWINDOW` Öffnet ein weiteres Textfenster.

`NEXT` Sucht in Vorwärtsrichtung nach dem Suchbegriff.

`NEXTR` Sucht in Vorwärtsrichtung nach dem Suchbegriff und ersetzt ihn.

`PAGEDOWN` Eine Seite abwärts.

`PAGESET` ⟨*N*⟩ Legt Seitengröße in N% der Zeilen eines Fensters fest.

`PAGEUP` Eine Seite aufwärts.

`PING` ⟨*N*⟩ Markiert eine Textstelle. Erlaubt sind Werte von 0 bis 9.

`PONG` ⟨*N*⟩ Springt zu einer markierten Textstelle.

`PREV` Sucht in Rückwärtsrichtung nach dem Suchbegriff.

© 1990 technicSupport

`PREVR` Sucht in Rückwärtsrichtung nach dem Suchbegriff und ersetzt ihn.

`QUIT` Das aktuelle Textfenster wird verlassen.

`RECALL` Wiederholt das vorherige Kommando. Hat keine Wirkung, wenn `RECALL` aus der Kommandozeile aufgerufen wird.

`REFORMAT` Formatiert einen Absatz.

`REMEOL` Löscht Text bis zum Zeilenende.

`REPSTR` ⟨*Text2*⟩ Bestimmt neuen Text für das Ersetzen.

`RESIZE` ⟨*Spaltenzahl*⟩ ⟨*Zeilenzahl*⟩
Ändert die Größe des Textfensters.

`RETURN` Wirkt wie die Befehlsfolge `FIRST DOWNADD`.

`RIGHT` Cursor um eine Position nach rechts.

`SAVEAS` ⟨*Name*⟩ Speichert Datei unter neu anzugebendem Namen.

`SAVEMAP` ⟨*Name*⟩
Speichert Tastaturbelegung in einer Datei.

`SAVEOLD` Speichert Datei unter dem bereits bestehenden Namen.

`SAVETABS` on|off Wandelt beim Speichern Leerzeichen um.

`SCREENBOTTOM` Springt in die letzte Zeile des Fensters.

`SCREENTOP` Springt in die erste Zeile des Fensters.

`SCROLLUP` Blättert aufwärts, ohne den Cursor zu beeinflussen.

`SCROLLDOWN` Blättert abwärts, ohne den Cursor zu beeinflussen.

`SETFONT` ⟨*Fontname*⟩ ⟨*Pixelgröße*⟩
Lädt einen neuen Zeichensatz

`SOURCE` ⟨*Name*⟩ Liest eine Scriptdatei ein.

`SPLIT` Trennt Zeile an der Cursorposition.

`TAB` Tabulatorsprung vorwärts.

`TABSTOP` ⟨*N*⟩ Setzt den Tabulatorabstand.

`TLATE` [+/-]⟨*N*⟩ Ändert den Wert eines Zeichens oder erlaubt die Eingabe eines absoluten Wertes für ein Zeichen.

`TOP` Sprung zum Textanfang.

`TOPEDGE` ⟨*N*⟩ Bestimmt die Anfangszeile eines neuen Fensters in Pixeln.

`UNBLOCK` Löscht Blockmarkierungen.

`UNMAP` ⟨*Tastenkombination*⟩
Löscht eine Tastenkombination.

`UP` Cursor eine Zeile aufwärts.

`WIDTH` ⟨*N*⟩ Bestimmt die Breite für neue Fenster in Pixeln.

`WLEFT` Springt an den Anfang des vorangehenden Wortes.

`WORDWRAP` on|off|toggle
Bestimmt, ob Wörter, die nicht mehr in die Zeile passen, automatisch in die nächste Zeile gezogen werden.

`WRIGHT` Springt an den Anfang des folgenden Wortes.

Die Abkürzungen für Sonderfunktionstasten

`bs`	Backspace-Taste
`tab`	Tabulator-Taste
`del`	Del
`help`	Help
`up`	↑
`down`	↓
`left`	←
`right`	→
`s`	⇧

© 1990 technicSupport

`c`	[Ctrl]
`a`	[Alt]
`A`	[Amiga]
`f1`	[F1] -Funktionstaste
`nk0`	[0] auf dem Ziffernblock
`L`	Linke Maustaste als Selektiertaste
`lmb`	Linke Maustaste als normale Taste
`R`	Rechte Maustaste als Selektiertaste
`rmb`	Rechte Maustaste als normale Taste

Kommandos für Fortgeschrittene

Neben den oben aufgeführten Standard-Kommandos kennt `Dme` noch einige komplexere Befehle, die die Entwicklung extrem leistungsfähiger Makros erlauben. Auf der anderen Seite kann eine Fehlprogrammierung solcher Makros aber auch schwerste Verwüstungen im Text anrichten. Ihre Funktionssicherheit sollte daher besonders gründlich ausgetestet werden, bevor sie auf wichtige Texte „losgelassen“ werden.

Zu diesen erweiterten Möglichkeiten zählt die Abfrage bestimmter Bedingungen, die automatische Wiederholung bestimmter Befehlsfolgen, die Ausgabe von Informationstexten aus anderen Dateien, die Übernahme des Wortes unter dem Cursor in eine Variable sowie das Aufrufen von `ARexx`-Kommandos.

An dieser Stelle ein paar Worte zu `ARexx`: Es ist die Amiga-Umsetzung der auf Großrechnern verwendeten Sprache `Rexx`. Eine besondere Fähigkeit dieser Sprache liegt darin, daß sie es erlaubt, Mitteilungen zwischen zwei ansonsten völlig selbständig laufenden Programmen auszutauschen. Bei mir genügt zum Beispiel ein Tastendruck im `CygnusEd`, um den momentanen Bearbeitungsstand einer Datei abzuspeichern, den Namen dieser Datei an TEX zu übermitteln und den Formatierprozeß zu starten.

Voraussetzung für diesen Datenaustausch ist das Vorhandensein einer **ARexx**-Schnittstelle in den entsprechenden Programmen. Während kommerzielle Programme wie TEX und **Ced**, aber auch das in letzter Zeit verstärkte Auftauchen von **ARexx**-Programmen und -Utilities in der PD, zeigen, daß diese Sprache in den USA immer beliebter wird, ist sie in Deutschland noch weitgehend unbekannt, auch ist **ARexx** nur mit englischer Dokumentation zu erhalten. Der Preis liegt mit etwa DM 100.- sehr günstig. Die Amiga-Umsetzung stammt übrigens von WILLIAM S. HAWES, der den meisten PD-Anwendern durch sein Shareware-Programm **Conman** bekannt sein dürfte. Es übrigens sehr wahrscheinlich, daß **ARexx** fester Bestandteil des Betriebssystems 1.4 werden wird.

Von diesen Fortgeschrittenenkommandos ist die Abfrage bestimmter Bedingungen besonders interessant. Zu diesem Zweck stellt **Dme** drei Befehle bereit.

if [!]⟨*Bedingung*⟩ ⟨*Befehl*⟩
: Der Befehl wird ausgeführt, wenn die Bedingung erfüllt ist. Wird der Bedingung ein „!“ vorangestellt, so wird der Befehl ausgeführt, wenn die Bedingung *nicht* erfüllt ist.

while [!] ⟨*Bedingung*⟩ ⟨*Befehl*⟩
: Der Befehl wird immer wieder ausgeführt, solange die Bedingung erfüllt ist.

ifelse [!] ⟨*Bedingung*⟩ ⟨*Befehl 1*⟩ ⟨*Befehl 2*⟩
: Befehl 1 wird ausgeführt, wenn die Bedingung erfüllt ist, ansonsten wird Befehl 2 gestartet.

Die Kürzel für die Bedingungen, die **Dme** berücksichtigen kann, sind in der folgenden Übersicht aufgeführt:

t	Cursor steht in erster Zeile.
b	Cursor steht in letzter Zeile.
l	Cursor steht am Zeilenanfang.
r	Cursor steht am Zeilenende.

© 1990 technicSupport

`x =|<|>`⟨N⟩	Die Spaltenposition des Cursors ist gleich, kleiner oder größer als ⟨N⟩.
`y =|<|>`⟨N⟩	Die Zeilenposition des Cursors ist gleich, kleiner oder größer als ⟨N⟩.
`cb`	Cursor steht innerhalb eines Blocks.
`m`	Text wurde geändert.
`i`	Einfügemodus ist aktiviert.
`cl`	Zeichen unter Cursor ist Kleinbuchstabe.
`cu`	Zeichen unter Cursor ist Großbuchstabe.
`ca`	Zeichen unter Cursor ist alphanumerisch.
`cn`	Zeichen unter Cursor ist numerisch.
`c =|<|>`⟨N⟩	Zeichen unter dem Cursor ist gleich, kleiner oder größer als der ASCII-Wert ⟨N⟩.

Zwei Beispiele sollen die Anwendung solcher Abfragen verdeutlichen. Als erstes soll durch Drücken von [F1] das Wort, auf dem sich der Cursor befindet, in Großbuchstaben umgewandelt werden:

```
map f1 (right wleft while c<>32 (if cl (tlate -32) right))
```

Zunächst wird der Cursor um eine Position nach rechts verschoben, da der `wleft`-Befehl zum *vorhergehenden* Wort springen würde, wenn sich der Cursor zufällig auf dem ersten Buchstaben des Wortes befände. Anschließend wandert der Cursor nach rechts, bis er auf ein Leerzeichen trifft (`while c<>32 ... right`). An jeder Position wird gefragt, ob der Cursor auf einem Kleinbuchstaben steht (`if cl`). Ist dies der Fall, so wird der ASCII-Wert des Zeichens in den Wert des entsprechenden Großbuchstabens umgewandelt (`tlate -32`).

Das zweite Beispiel stammt unmittelbar aus der Arbeit an den PD-Büchern. Da die Tastatur keine Anführungszeichen kennt, die den Normen des Schriftsatzes entsprechen („“), werden sie mit Hilfe eines Makros verwirklicht. Ein Satz wie

„Wie spät ist es denn“, fragte Peter.

sieht in meiner TEX-Datei folgendermaßen aus:

```
\sag{Wie spät ist es denn}, fragte Peter.
```

In den unformatierten Rohtexten stehen allerdings die gewöhnlichen Anführungszeichen

```
"Wie spät ist es denn", fragte Peter.
```

Das folgende Makro — in der „.edrc"-Datei steht es in einer Zeile — ersetzt diese Zeichen automatisch durch das entsprechende TEX-Makro, wenn [F5] gedrückt wird. Es funktioniert nur dann ordnungsgemäß, wenn der Einfügemodus aktiviert ist. Bei der Cursorposition beginnend, wird nach dem ersten Vorkommen eines Anführungszeichens (Wert 34) gesucht und dieses durch `\sag{` ersetzt. Das zweite gefundene Anführungszeichen wird gegen `}` ausgetauscht. Die Befehlsfolge

```
(ifelse x=80 (down first)
```

sorgt dafür, daß der Cursor am Ende einer Zeile, hier definiert als Spalte 80, zum Anfang der folgenden Zeile springt. Hier das komplette Makro:

```
map f5 (while c<>34 (ifelse x=80 (down first) right) tlate 92 right
(sag{) while c<>34 (ifelse x=80 (down first) right) del (}))
```

PROBLEME

Wird `Dme` ohne Angabe einer zu bearbeitenden Datei aufgerufen und anschließend ein Arp-Dateiauswahlrequester geöffnet, dann wird der provisorische Dateiname „Unnamed" fehlerhaft übergeben, und der Requester meldet einen „Directory Error". In diesem Fall empfiehlt es sich, mit der rechten Maustaste in die Titelleiste des Dateiauswahl-Fensters zu klicken. Der Requester zeigt dann alle Verzeichnisse an, die jetzt durch einfaches Anklicken angezeigt werden können.

© 1990 technicSupport

HERKUNFT

Die besprochene Version 1.30 findet sich auf der Fishdisk 153. Autor ist:

```
Matthew Dillon
891 Regal Rd
Berkeley, California  94708
USA
```

VERWEISE

`CedDemo` (RPD 040), `MG2b` (Fishdisk 147), `Uedit` (Fishdisk 121), `MEd` (RPD 051)

ExplodingWindows

Effektvolles Öffnen von Fenstern

AUFRUF

Das Programm kann durch Anklicken seines Icons gestartet werden. Ein erneutes Anklicken entfernt das Programm wieder. Aus dem CLI heraus lautet der Aufruf:

run[back] EW [-F0] [-F1] [-L0] [-L1] [-RG] [-RP] [-S0] [-S1]

Damit das CLI-Fenster wieder geschlossen werden kann, sollte man statt `run` besser den `Runback`-Befehl (s. u.) verwenden.

OPTIONEN

`-F0 bis -RP`	Diese Optionen bestimmen Art und Verlauf der Linien und können kombiniert werden. Ihre optische Wirkung läßt sich schwer beschreiben; am besten probiert man sie aus.
`-S1`	Unterdrückt das Fenster mit der Startmeldung.
`-S0`	Läßt das Fenster mit der Startmeldung erscheinen.

ZWECK

`ExplodingWindows` sorgt beim Öffnen und Schließen von Fenstern für einen attraktiven optischen Effekt.

BESCHREIBUNG

Nach dem Starten des Programmes von der Workbench zeigt sich für kurze Zeit ein Fenster mit der Mitteilung, daß `EW` nun installiert ist. Von nun an zeigt sich beim Öffnen aller Fenster ein netter Effekt: Ein Netz aus feinen Linien dehnt sich über der Fläche aus, die das Fenster einnehmen wird. Erst dann erscheint das Fenster selbst. Beim Schließen eines Fensters zieht sich das Netz entsprechend zusammen. Das ganze geht so schnell, daß kein großer Zeitverlust entsteht. Möchte man diesen Effekt auf Dauer installieren, sollte man

© 1990 technicSupport

das Programm mit seinem Icon aus der `Graphics`-Schublade der TBAG-Disk in die `System`-Schublade seiner Arbeitsworkbench ziehen, d. h. das Bildsymbol anklicken, die linke Maustaste gedrückt halten, das Symbol in die Zielschublade ziehen und dort erst die Maustaste loslassen. Dann wird nur noch der `Runback`-Befehl (z. B. auf Fishdisk 152) benötigt, der in das `C`-Verzeichnis der Workbench kopiert werden muß:

```
copy AmigaLibDisk152:runback/runback c:
```

In die `Startup-Sequence` der Workbench ist die Befehlszeile

```
runback sys:system/ew
```

einzufügen, die den automatischen Aufruf des Programmes beim Starten des Amiga mit dieser Diskette bewirkt.

HERKUNFT

Die hier besprochene Version 1.4 des Programmes befindet sich auf der TBAG-Diskette 28. Die Adresse des Autors:

`Ed Scherer, PLink ID: Syn*Ed, CIS ID: 73457,151`
`15220 Dearborn`
`Overland Park, KS 66223`

Flaschbier

Das Werner-Spiel

AUFRUF

Sie starten das Spiel am besten, indem Sie den Rechner nicht mit der Workbench, sondern mit der Werner-Disk booten. Die zunächst erscheinenden beiden Intros und das Titelbild beenden Sie mit der linken Maustaste. Sie erreichen jetzt ein Menü, in dem Sie das Spiel mit [F1] starten können.

ZWECK

`Flaschbier` ist ein schnelles Boulder-Dash-Spiel mit guter Grafik.

BESCHREIBUNG

Die meisten kennen sicher die Kult-Comic-Figur Werner aus dem hohen Norden Deutschlands, genauer gesagt aus Flensburg. Neben vielen grünen Wiesen gibt es hier auch das allseits beliebte Flaschenbier, von dem auch Werner gern mal'n Schluck nimmt.

Im Spiel geht es darum, daß Sie die Rolle Werners übernehmen und eine Flasche des „edlen Flensburger Pilseners" (auch *„Flens"* genannt) finden. Die Arbeit wird Ihnen allerdings durch drei Dinge erschwert. Zunächst ist da die Zeit: Sie haben genau 100 Sekunden. Das wäre nicht weiter tragisch, wenn es nicht die eifrigen Polizisten gäbe, die einen ganz schön ins Schwitzen bringen können. Als dritte Hürde gibt es noch die Steine, vor denen man sich hüten sollte. Fallen sie Werner auf seinen Brummschädel (bei so viel *„Flens"* kein Wunder), so tut ihm das gar nicht gut. Das ist auch schon in groben Zügen der Inhalt des Spiels.

Falls Sie jetzt glauben, das Spiel sei ziemlich einfach und langweilig, haben Sie sich gewaltig getäuscht. Um mehr als die ersten paar Spielstufen erfolgreich zu bestehen, erfordert es viel Überlegung und Geschick. Wichtig ist hierbei vor allem, daß Sie genau wissen, wie sich die Polizisten verhalten und wie die Steine fallen. Hierfür gibt es bestimmte Regeln, die ich Ihnen kurz erklären möchte.

© 1990 technicSupport

Die Polizisten laufen solange geradeaus, bis sie auf ein Hindernis stoßen. Hindernisse können Steine, Dreck oder eine Wand sein. Sie versuchen dann, zunächst nach rechts auszuweichen, ist dies nicht möglich, laufen sie nach links. Nur wenn beide Richtungen durch ein Hindernis versperrt sind, kehren sie um.

Die Steine fallen solange herunter, bis sie auf ein Hindernis stoßen. Ist das Hindernis Dreck oder eine Wand, so bleiben sie dort liegen. Ist es aber ein Stein, so fallen sie, falls dort Platz ist, rechts daneben, sonst nach links.

Das Spiel

Vor Beginn eines jeden Levels, haben Sie Zeit, sich einen möglichst günstigen und vor allem schnellen Weg zum verdienten kühlen Naß zu suchen. Es geht bei diesem Spiel nicht nur um flinke Joystick-Akrobatik, sondern auch um Köpfchen. Es kann sehr hilfreich sein, sich zunächst einige Positionen zu suchen, auf denen man Halt machen kann, ohne von Polizisten überrascht zu werden. Manchmal kann man Polizisten aber auch in die Irre leiten, indem man ihnen ein Hindernis aus dem Weg räumt. Dann rennen sie irgendwo auf dem Bildschirm herum, während man selbst ohne allzuviel Hektik an das begehrte Objekt gelangen kann.

Erst durch Drücken des Feuerknopfes am Joystick starten Sie die jeweilige Spielstufe. Ab jetzt haben Sie 100 Sekunden, um zur Flasche zu gelangen. Haben Sie dieses Ziel erreicht, wird die verbleibende Zeit zu den Scores (Punkten) addiert.

Sie starten mit zehn Leben, und für jede erfolgreich bestandene Spielstufe erhalten Sie eines dazu. Werden Sie von einem Polizisten erwischt, von einem Stein auf den Kopf getroffen oder überschreiten die Zeit, so verlieren Sie ein Leben und müssen den Level von vorne beginnen.

Mit [F1] können Sie den momentanen Level abspeichern und mit [F2] wieder laden. Bedenken Sie jedoch, daß Sie nur den Level, nicht aber die Scores (Punktestand) abspeichern können. Nach dem Laden eines zuvor abgespeicherten Spiels beginnen Sie immer mit

0 Punkten. Möchten Sie eine Pause einlegen, drücken Sie einfach [F1] oder [F2] , ohne danach einen Namen einzugeben. Zum Weiterspielen drücken Sie nur [Return] , der Punktestand bleibt hierbei erhalten. Sollten Sie einmal gefangen sein und nicht mehr herauskommen, so müssen Sie nicht warten, bis die Zeit abgelaufen ist. Sie können den Level mit [Space] beenden. Sie verlieren dann ein Leben und fangen den Level noch einmal von vorne an. Mit [Escape] können Sie das Spiel beenden.

Übrigens, es soll auch einen Trainermodus geben. Fragt sich nur, wie man dort hineinkommt. Mir ist es nicht gelungen, aber vielleicht gibt es ja den einen oder anderen Freak, der mit viel Fantasie und Geduld die richtige Tastenfolge am richtigen Ort findet. Falls Sie Erfolg haben, lassen Sie es mich wissen, ich würde mich sehr darüber freuen.

Der Wettbewerb

Die Autoren des Spiels haben einen Wettbewerb ausgeschrieben. Wer als erster alle 250 Level (ja so viele gibt es wirklich!) durchspielt, gewinnt 10 Leerdisketten. (Anm. d. Hrsg.: Immerhin besser als zehn leere Flaschen Flens!) Als Nachweis schicken Sie einfach die Zahl, die nach dem 250 Level erscheint an die nachstehende Adresse:

```
PLK 106138 C
6750 Kaiserslautern
West-Germany
```

Damit niemand auf die Idee kommt, in einem abgespeicherten Spiel die Levelnummer mit einem Disk-Monitor zu ändern, erscheint die korrekte Zahl nur, wenn alle 250 Level in einem Zug durchgespielt werden, ohne zwischendurch abzuspeichern. Sie sollten sich also etwas Zeit nehmen und sich mit ausreichend Nahrungsmitteln ('ne Kiste *„Flens“* tuts auch) eindecken, falls Sie vorhaben, die 10 Leerdisketten zu gewinnen.

Warnung:
Es könnte eine Art Sucht entstehen, die Sie vielleicht schon von anderen Boulder-Dash-Spielen wie `Emerald Mine` kennen. Das Werner Spiel `Flaschbier`, das schon auf dem C64 viele fesselte, ist eine sehr

 © 1990 technicSupport

gelungene Umsetzung. Und dazu noch Public Domain: Sie können es an Ihre Freunde weitergeben, ohne dabei ein schlechtes Gewissen haben zu müssen.

PROBLEME

Auf meinem Amiga mit 3 MByte lief das Spiel einwandfrei, vorausgesetzt alle externen Laufwerke waren abgeschaltet. Bei Rechnern mit nur 512 KByte kann es allerdings zu Speicherproblemen kommen. Sie sollten dann, neben dem Abschalten der externen Laufwerke, auch darauf achten, daß Sie keine resetfesten Programme im Speicher halten. Außerdem empfiehlt es sich, direkt nach dem Erscheinen des Hauptmenüs, mit [F1] das Spiel zu starten, ohne vorher irgendwelche anderen Dinge auszuwählen. Es könnte sonst, ebenfalls wegen Speichermangels, zu einem Absturz kommen.

HERKUNFT

Das Werner Spiel `Flaschbier` befindet sich auf der ACS 141. Geschrieben wurde es von einer bekannten Crackergruppe, weshalb die Autoren auch nur mit Ihren „Künstlernamen“ vermerkt sind. Es zeigt mal wieder, daß Cracker, die häufig nur Kopierschutze entfernen oder Intros schreiben (zwar ganz lustig anzusehen, aber sonst ziemlich langweilig) durchaus in der Lage sind, ein interessantes Spiel zu programmieren. `Flaschbier` ist Public Domain, trotzdem bittet der Autor um eine kleine Spende von DM 10,-, sollte Ihnen das Spiel gefallen. Da der Autor anonym bleiben möchte, können Sie das Geld an die gleiche Adresse schicken, die auch für den Wettbewerb gilt.

FONTCON

Wandelt Amiga-Zeichensätze in Druckerfonts

AUFRUF

`FontCon` läßt sich von der Workbench aus starten. Im CLI wird das Programm durch die Eingabe seines Namens aufgerufen.

ZWECK

`FontCon` ermöglicht bei entsprechend ausgerüsteten Druckermodellen die Verwendung neuer Zeichensätze anstelle der druckereigenen Fonts.

BESCHREIBUNG

Verschiedenartigste Zeichensätze für den Bildschirm sind schon lange keine Mangelware mehr. In den `fonts`-Verzeichnissen von Workbench- und Extras-Diskette liegen die bekannten Fonts, die von der seriösen Helvetica bis zur verspielten Sapphire-Schrift reichen; auf Public Domain Disketten findet man eine Riesenauswahl neuer Schrifttypen, aber auch Sonderzeichensätze, die beispielsweise das Verarbeiten von kyrillischer oder Esperanto-Schrift ermöglichen; mit dem PD-Programm `TexF` (s. Das Zweite Amiga Public Domain Buch, S. 256–261) lassen sich `pk`.-Fonts in Amiga-Zeichensätze konvertieren. Und schließlich gibt es diverse Editoren, mit denen sich Zeichen ganz nach eigenem Geschmack erstellen lassen.

Auch wenn manche dieser Zeichensätze auf dem Bildschirm ganz nett aussehen mögen, können sie doch nur dann von echtem Nutzen sein, wenn sie auch auf dem Drucker ausgegeben werden können. Doch gerade hier beginnen die Schwierigkeiten. Manche Textprogramme können die Amiga-Zeichensätze überhaupt nicht nutzen, andere bringen sie lediglich durch eine Bildschirm-Hardcopy zu Papier. Bei diesem Verfahren gibt der Drucker im Grafikmodus die Pixelinformationen des Bildschirms wieder, was recht zeitaufwendig ist, aber dennoch zu keiner befriedigenden Ausgabequalität führt. Mit dem `Notepad` (`Print As Graphic`-Menü) kann diese Methode leicht ausprobiert werden.

 © 1990 technicSupport

Viele moderne Drucker geben dem Anwender jedoch die Möglichkeit, selbstdefinierte Zeichensätze in den Drucker zu laden, und diesen Zeichensatz anstelle der druckereigenen Fonts zu verwenden. Nur fehlte für den Amiga bislang ein Programm, das eine solche Aufgabe übernehmen kann, weshalb WERNER GÖRTZ es sich nicht nehmen ließ, in seiner Beschreibung des Zeichensatzeditors `FontEdit` (im Zweiten Amiga PD-Buch, S. 209–215) unsere Leser zum Schreiben eines solchen Programmes anzuregen. Dieser Aufruf kam ein wenig spät, denn TIM PIETZCKER hatte bereits ein derartiges Projekt begonnen. Der Aufruf veranlaßte den Autor jedoch, uns das fertige Programm `FontCon` zuzuschicken, wofür wir uns an dieser Stelle noch einmal herzlich bedanken. Wenn die eigentliche Programmbeschreibung an dieser Stelle sehr kurz ausfällt, dann liegt dies ganz einfach daran, daß das Programm bereits mit einer sehr ausführlichen deutschen Anleitung versehen ist.

`FontCon` ist für all diejenigen geeignet, die über einen NEC P6 oder einen zu diesem kompatiblen Drucker verfügen. Bei Druckern, die die Form selbstdefinierter Zeichensätze in anderer Weise bestimmen, müßte der mitgelieferte C-Code des Programmes entsprechend angepaßt werden.

`FontCon` läßt sich sehr komfortabel bedienen. Alle Zeichensätze, die sich im logischen `Fonts:`-Gerät befinden, können durch Angabe ihres Namens und ihrer Größe geladen werden. Bevor man durch einfaches Anklicken des [Start!] -Buttons die Umwandlung in einen Druckerfont startet, kann man festlegen, ob die Ausgabe direkt an den Drucker geschickt oder in eine Datei geleitet werden soll. Sehr kleine Zeichensätze lassen sich gleich bei der Umwandlung vergrößern. Ist der neue Zeichensatz zum Drucker geschickt worden, kann er mit [Teste den Font] sofort ausprobiert werden.

Wird der Zeichensatz in eine Datei geleitet, so kann er jederzeit auch ohne Benutzung von `FontCon` aktiviert werden. Dazu ist die entsprechende Datei lediglich an die Druckerschnittstelle zu schicken:

`copy` ⟨*Dateiname*⟩ `to par:`

An dieser Stelle ist es sehr wichtig, daß die Datei an eine der physikalischen Schnittstellen (`par:` oder `Ser:`) geschickt wird, nicht

FontCon wandelt Amiga-Zeichensätze in Druckerfonts

aber an das `prt:`-Gerät, da die druckereigenen Steuersequenzen, die für das Laden eines selbstdefinierten Zeichensatzes zuständig sind, vom `prt:`-Gerät nicht verarbeitet werden können. Die nachfolgenden Textdateien können aber selbstverständlich wieder an `prt:` geschickt werden.

BEISPIELE

Möchte man den Helvetica-Zeichensatz der `Extras1.3D`-Diskette in einen Druckerfont umwandeln, muß zunächst einmal das `Fonts:`-Gerät neu zugewiesen werden:

```
assign fonts: Extras1.3D:fonts
```

Anschließend startet man `FontCon`, trägt in die entsprechenden Gadgets den Namen (`Helvetica`) und die Größe (`24`) ein und klickt das [Lade Font!] -Gadget an. Man wählt einen Namen wie `Ram:Helv24` für die Ausgabedatei und klickt auf das [Datei] -Feld, sowie auf [Printer] . Die Umwandlung wird dann mit [Start !] veranlaßt.

 © 1990 technicSupport

Nach dem Verlassen des Programms begibt man sich wieder in das CLI, wo man zunächst das Laden des `printer.device` veranlaßt:

```
echo >prt: "" noline
```

Besitzt man die Workbench 1.3 nicht, muß man die `noline`-Option weglassen; der Drucker führt dann aber einen Zeilenvorschub aus. Jetzt wird mit

```
copy Ram:Helv24 par:
```

der neu geschaffene Font an den Drucker geschickt. Mit Hilfe des Programmes `NECquik` (ebenfalls in diesem Buch besprochen) können die Zeichen auf Wunsch vergrößert werden. Nach diesen Vorbereitungen werden Textdateien, die mit

`copy` ⟨*Textdatei*⟩ `prt:`

an den Drucker geschickt werden, im neuen Zeichensatz ausgedruckt.

PROBLEME

Vor dem Laden eines benutzerdefinierten Zeichensatzes in den Drucker sollte das `prt:`-Gerät wenigstens einmal angesprochen werden, z. B. mit dem Befehl:

```
echo >prt "" noline
```

Hierdurch wird das Laden des `printer.device` veranlaßt, was mit einem Setzen der Druckereinstellungen auf die Standard-Werte verbunden ist. Erfolgt der erste Aufruf des `prt:`-Gerätes nach dem Einladen des selbstdefinierten Zeichensatzes, so wird der Drucker automatisch wieder auf seinen internen Zeichensatz zurückgestellt. In diesem Fall kann die Font-Auswahltaste des NEC-Druckers solange betätigt werden, bis die LED-Anzeige auf „A“ steht; es kann jedoch auch das Programm `NECquik` benutzt werden, um den selbstdefinierten Zeichensatz anzuwählen.

In ihrer Grundgröße sind die mit `FontCon` erzeugten Druckerzeichensätze sehr winzig. Diesem Manko kann durch Vervielfachen der

Zeichenhöhe bzw. -breite abgeholfen werden. Am einfachsten läßt sich dies ebenfalls mit dem Programm `NECquik` verwirklichen.

`FontCon` erzeugt Proportionalschrift, bei der die einzelnen Zeichen eine unterschiedliche Breite aufweisen können. Ein Text der im Editor oder im Textprogramm im Blocksatz gesetzt ist, wird daher beim Ausdruck einen Flatterrand erhalten. Man kann die Formatierung jedoch auch dem Drucker überlassen, indem man mit `NECquik` die Ränder auf die gewünschten Werte stellt und anschließend die hexadezimale Sequenz `1B 61 03` sendet. Bei diesem Verfahren muß man allerdings damit rechnen, daß der Zeilenumbruch nicht in jedem Fall optimal arrangiert wird. Es kann schon mal vorkommen, daß ein einzelnes Wort in einer Zeile erscheint, weil in der vorhergehenden nicht mehr genügend Platz war.

HERKUNFT

Die hier besprochene Version 1.1D findet man mit dem C-Quellcode auf der Franz 45. Die Adresse des Autors:

`Tim Pietzcker`
`Schwarzwaldstr.52`
`D-7819 Denzlingen`

 © 1990 technicSupport

Freud

Ein höchst interessanter Persönlichkeitstest

AUFRUF

Das Programm **Freud** ist in Basic geschrieben, daher benötigt es die „Extras"-Diskette mit dem **AmigaBasic**-Interpreter. Wesentlich ist, daß *dieser* Diskettenname während des Starts von **Freud** abgefragt wird, was bedeutet, daß man den Namen der deutschen Version von „ExtrasD" in „Extras" ändern muß. Also zieht man am besten eine Kopie seiner Diskette und benennt diese Kopie über das **Workbench**-Menü (Untermenü **Rename**) in „Extras" um. **Freud** selbst befindet sich im Unterverzeichnis **BasicProgs** der Fishdisk. Von dort läßt es sich mit einem Doppelklick auf sein Icon starten.

ZWECK

Den Zweck dieses *Multiple-Choice-Tests* nennt schon der Autor: Man erspart sich eine zwölfjährige Psychotherapie und $50.000. Aber hält dieser Test auch, was der Autor verspricht? Ich denke ja, zumindest regt er zum Nachdenken an und ist sehr amüsant. Ach, übrigens — es hat nur für Männer Sinn, diesen Test mit seinen 48 Fragen zu machen, denn die Fragen und die anschließende Analyse sind nur auf Männer ausgerichtet. Warum? Nun, der Psychoanalytiker Freud verstand die Frauen nicht, und der Programmierer hat sich dieser Einstellung angeschlossen. Wenn Frauen diesen Test trotzdem machen, gilt hier die Emanzipation — sie werden genauso behandelt wie Männer!

BESCHREIBUNG

Schon das Lesen des Dokumentartextes verspricht Interessantes, fordert doch Programmierer George Trepal den Leser dramatisch dazu auf, nicht weiterzulesen: Das Testergebnis würde sich verfälschen. Um also keinen Einfluß zu nehmen, soll hier auch nicht weiter darauf eingegangen werden. Nur so viel: Die Testteilnehmer werden in drei Klassifikationen unterteilt:

© 1990 technicSupport

1. Der Phallus-Typ
2. Der orale Typ
3. Der anale Typ

Auch auf die Einzelheiten der Testfragen möchte ich mich hier nicht näher einlassen, aber sie decken wirklich fast alle Lebensbereiche ab — vom Sex bis zu allgemeinen Alltagsfragen. Wirklich — wer bei der Beantwortung schummelt, beschummelt nur sich selbst. Natürlich kann man Fragen auch so beantworten, wie nach der eigenen Meinung ein Freund sie beantworten würde, aber die Antworten werden halt nur die eigene Meinung über den Freund wiedergeben.

Für alle, die nicht „so fit im Englischen" sind, möchte ich mir die Mühe machen, die Erklärungen zum jeweiligen Typ zu übersetzen. Diese Erklärungen stehen in der Datei Doc.Freud (das Piktogramm gefällt mir sehr gut). Wer will, braucht also den folgenden Text nicht zu lesen. Aber auch für die anderen Leser gilt: Bitte überspringen Sie die folgenden Abschnitte und gehen Sie zum Abschnitt „Durchführung des Pesönlichkeitstests" über! Lesen Sie bitte — um Manipulationen zu vermeiden — die folgenden Kapitel erst nach Beendigung des Testes!

STOP — NOCH NICHT WEITERLESEN — STOP

Psychoanalytische Grundlagen

Der F1-Typ (der Don Juan oder auch Phallus-Typ):

Dieser Typ glaubt diesem Test am wenigsten — er wird nämlich mit teilweise unangenehmen Wahrheiten konfrontiert. Der Don Juan ist der nach außen Orientierte, er ist extrovertiert. Seit seiner frühesten Jugend ist er darauf trainiert, daß er bei anderen nur akzeptiert wird, wenn er besonders charmant und entgegenkommend

© 1990 technicSupport

ist. So hat er gelernt, nur nach außen zu wirken und sich selbst zu vernachlässigen. Da er immer auf äußere Wirkungen fixiert ist, gehören auch Bücher über Sexualpraktiken zu seiner hauptsächlichen Lektüre. Er ist immer auf der Suche nach Akzeptanz — und wird er nicht akzeptiert, so baut er sich schließlich seine eigene Scheinwelt auf. Wird der Phallus-Typ aber von seinem sozialen Umfeld akzeptiert und kann er sich frei entfalten, dann ist er für seine Umwelt oft ein gutes Vorbild. Er ist dann ein guter Familienvater und für seine Kinder eine Leitfigur. Wenn er sich von jemandem akzeptiert fühlt, dann tut er alles für ihn — auch um sich die Zuneigung weiter zu erhalten.

Der F1-Typ braucht sofortige Reaktion, mit einer abwartenden oder zurückhaltenden Haltung kann er wenig anfangen. Neue Sexualpartner machen ihn immer neugierig — vor allem *seine* Wirkung auf sie. Diesbezüglich ist er sozusagen „immer auf der Jagd". Der Don Juan sucht darin immer eine neue Selbstbestätigung. Beruflich ist er sehr engagiert und erfolgreich — vorausgesetzt, daß er dem Chef wichtig ist und seine Mitarbeiter ihn anerkennen. Er fürchtet die Kritik nicht, ist aber auch sehr weit davon entfernt, sie zu akzeptieren. Sein Fehler: Bevor er schlechte Neuigkeiten akzeptiert, flüchtet er sich in Tagträume.

Nun aber zur psychologischen Komponente dieses Tests: Der F1-Typ gibt als letzter zu, daß er psychische Probleme hat. Er ist daher sehr empfänglich für die *Midlife-crisis*; dann nämlich wird er wegen psychosomatischer Erkrankungen vergeblich Ärzte aufsuchen. Vergeblich deshalb, weil er als extrem extrovertierter Typ nie gelernt hat, auf sein Inneres zu hören. Jetzt aber müßte er nach innen schauen — und dadurch geriete sein Weltbild in's Wanken.

Da der F1-Typ der extrovertierte Typ ist, nimmt er auch in der Mode eine Vorreiterrolle ein. Er will immer im Mittelpunkt stehen und glänzen. Für ihn ist es wichtig, daß er „in" ist — in allen Lebensfragen. Seine Hauptangst ist es, nicht der Beste zu sein. Eine seiner „Lebensweisheiten" ist: „Es ist nicht genug, Erfolg zu haben, andere müssen auch Mißerfolg haben" (DAVID MERRICK).

© 1990 technicSupport

Der F3-Typ (der orale Typ):

Dieser Typ akzeptiert diesen Test am ehesten, vor allem aber die negativen Resultate. Er ist der geborene Pessimist und immer auf der Suche nach Liebe und Zuneigung. Der orale Typ wird als solcher bezeichnet, weil er immer etwas im Mund zu haben scheint: Essen, Zigaretten, Kugelschreiber, Bleistifte u.s.w. Seine Mutter hat ihn immer vernachlässigt und zurückgewiesen. So ist er noch immer auf der Suche nach einer solchen. Eine Frau, die dem F1-Typ nahe kommt, ist eine optimale Partnerin für ihn, denn der orale Typ sucht eine starke Partnerin.

Im Gegensatz zum Don Juan ist der F3-Typ der introvertierte Typus. Der wichtigste Aspekt beim Geschlechtsverkehr ist für ihn, daß sich seine Sexualpartnerin um ihn kümmert, alles andere ist eher unwichtig für ihn. Normalerweise hat der orale Typ keine Potenzprobleme, aber er fürchtet sich vor Impotenz — er könnte sie ja bekommen.

Seine Lebenseinstellung ist pessimistisch. Sagt eine Geliebte zu ihm, daß er ein ausgezeichneter Liebhaber sei, dann glaubt er, daß sie ihn mit der Wahrheit nicht verletzen will. Sagt sie hingegen, daß er schlecht war, bestätigen ihn diese Worte nur in seiner negativen Selbsteinschätzung. Hat der F1-Typ die Überzeugung, daß seine Umwelt an allem die Schuld trägt, so sucht der F3-Typ die Fehler immer bei sich selbst und ergeht sich in quälerischen Selbstzweifeln. Überhaupt zeigt der orale Typ diverse Ansätze zum Masochismus.

Erstaunlicherweise ist seine Schwäche seine Stärke. Er vermittelt seinen Mitmenschen den Eindruck, daß sie Rücksicht auf ihn nehmen müssen und daß er doch soviel Zuneigung braucht — erstaunlicherweise wird meistens darauf eingegangen. Wird der F1-Typ mit Widerstand konfrontiert, versucht er ihn zu überwinden. Der F3-Typ dagegen glaubt schon von Anfang an, daß er verlieren wird...und verliert! So gehört es zur Grundhaltung des oralen Typs, Probleme zu vermeiden. Gerät er trotzdem in schwierige Situationen, agiert er nicht, sondern „sitzt sie aus". (Anm. des Hrsg.: Gewisse Ähnlichkeiten mit einer sehr prominenten Person sind rein zufällig.) Da der F3-Typ sehr viel Unterstützung und Zuneigung braucht, sind Freunde

© 1990 technicSupport

und Freundschaften sehr wichtig für ihn. Aber er braucht soviel Zuneigung und Sympathie, daß er immer zuwenig davon zu bekommen scheint. Durch die schlechten Erfahrungen mit seiner Mutter denkt er immer, daß die ihm Zuteil gewordene zwischenmenschliche Wärme mit einem Mal wie eine Seifenblase zerplatzen wird.

In der Ehe mit dem F3-Typ führt meistens die Frau das Regiment, im Bett und außerhalb. Am besten harmoniert er in einer Partnerschaft, in der der eine Partner die Kindes- und der andere die Erwachsenenrolle übernimmt. Solange nicht beide die Kindesrolle übernehmen, geht es gut. Meistens scheut sich der F3-Typ vor der Konfrontation mit der Realität. Stattdessen belastet er sich allzuoft mit vor allem psychischen Problemen, ohne sich aber richtig mit diesen oder sich selbst auseinanderzusetzen. Falls jemand ein großes Ziel auf seine Fahnen schreibt, wird er im F3-Typ einen begeisterungsfähigen Anhänger finden, denn der F3-Typ sucht Leitfiguren. Der F3-Typ als Leser wird über diese doch sehr negative Beschreibung nicht begeistert sein, sie aber gerade deswegen bereitwillig als Tatsache hinnehmen.

Der F5-Typ (die anale Persönlichkeit):

Dieser Typ akzeptiert alle Teile dieser „Analyse“ bis auf den Teil, der sich auf seinen Charakter bezieht. Anale Persönlichkeiten zeichnen sich hauptsächlich dadurch aus, daß sie im Vergleich zu den beiden anderen Typen einen starren und autoritären Charakter haben. Der F5-Typ ist von seiner Lebensweise hundertfünfzigprozentig überzeugt und hätte es am liebsten, wenn sich alle anderen seine — auch seine moralischen — Ansichten zu eigen machen würden. Ihm sind Entscheidungen am liebsten, in denen es nur ein „Ja“ oder ein „Nein“ gibt. Aber hier ist er auch in der Lage, unpopuläre Entscheidungen zu treffen, zumal er ja davon überzeugt ist, moralisch völlig im Recht zu sein. Dadurch kann er aber bei günstigen Konstellationen auch sehr viel zum Wohle anderer tun, denn er paßt sich nicht irgendwelchen Strömungen an.

Im Berufsleben ist die anale Persönlichkeit eher in den wissenschaftlichen und buchhälterischen Tätigkeiten anzutreffen. Sie eignet sich

© 1990 technicSupport

hervorragend zum Polizeibeamten, aber auch zum Prediger. In der Weitergabe seiner Überzeugung neigt der F5-Typ leicht zum Fanatismus. Die Eltern wirken im F5-Typ weiter, sie waren ihm gegenüber immer sehr autoritär. In seiner Kindheit wirkte er immer sehr erwachsen, aber nur, weil er weniger spontan und viel zurückhaltender als andere Kinder war. Als Erwachsener möchte er allen anderen am liebsten vorschreiben, wie sie zu sein haben. In der Art und Weise, anderen seine Lebensweise nahezubringen, kann der F5-Typ sehr unmoralisch vorgehen. Er sieht überall unmoralische Zustände, über die er sich dann ereifert. Aber diese unmoralischen Zustände sind nur eine Projektion seines von ihm nicht akzeptierten Sexualtriebes. Der F5-Typ ist ein guter Ausführer und Verwalter, weniger gut ist er, wenn Kreativität und Spontaneität gefragt sind, dazu ist er zu unflexibel. Die beiden anderen Typen versteht der F5-Typ überhaupt nicht. Der Don Juan ist ihm zu impulsiv und zu wenig selbstbeherrscht, der orale Typ zu zurückhaltend und zögerlich. Am liebsten würde er beide Typen strafen und dazu zwingen, so zu werden, wie er selbst ist. So sorgt der anale Typ am Arbeitsplatz dafür, daß der Don Juan eingeschränkt und dem schwächlichen F3-Typ gekündigt wird.

Nur wenn er Ereignisse und Tätigkeiten mit dem Verstand verarbeiten kann, findet der F5-Typ Gefallen an ihnen. So wirkt er oft verklemmt, was er durch Zurückhaltung zu verbergen versucht. Genauso ergeht es ihm auch mit seiner Sexualität. Impotenz kommt bei ihm nicht gerade selten vor. Eine frigide Frau belastet ihn daher nicht allzusehr. Aber gerade Frauen, die eher dem oralen Typus entsprechen, „fliegen" auf ihn, hat er doch für sie den Reiz einer Vaterfigur. Da aber der anale Typ von anderen erwartet, daß sie „erwachsen" sind, ist solch eine Beziehung selten von Dauer. Auch der F5-Typ ist sehr empfänglich für die *Midlife-Crisis*. In diesem Alter erkennt er oft, daß er sich selbst in einem Gefängnis eingeschlossen hat. Entweder lebt er dann sein Leben weiter, oder aber er verändert es. Da der F5-Typ zu radikalen Entscheidungen neigt, kann es vorkommen, daß er sein Vorleben — auch seine Ehe — über Bord schmeißt. Der F1-Typ verdrängt alle Schwächen, der F3-Typ steigert sich geradezu in diese hinein, der F5-Typ dagegen versucht sich mit diesen auseinanderzusetzen. Dabei bekämpft er auch sich

© 1990 technicSupport

selbst, denn er will nicht akzeptieren, daß er auch menschliche Seiten hat. Menschliche Schwächen gesteht er weder sich noch anderen zu.

Einschränkungen:

Man sollte die Charakterisierung der einzelnen Typen nicht zu ernst nehmen. In der Regel wird sich sowieso herausstellen, daß man ein Mischtyp ist. Also zeigt sich meistens nur eine Tendenz in eine oder mehrere Richtungen.

Durchführung des Persönlichkeitstests

Nachdem das `Freud`-Programm durch einen "Doppelklick" auf das Piktogramm „Freud.Loader" gestartet wurde, muß man durch Betätigen der Funktionstasten [F1], [F3] oder [F5] (auch Anklicken mit der linken Maustaste ist möglich) die Fragen beantworten. In dem Programm ist auch eine `Help`-Funktion integriert, die ebenfalls durch Anklicken selektiert werden kann. Im `Help`-Menü gibt es vier Alternativen, die durch die Eingabe der jeweiligen Buchstaben gewählt werden:

`S` Frage überspringen

`B` Eine Frage zurück und Chance zur Neubeantwortung

`F` Unbeantwortete Fragen finden (es erscheint eine Liste)

`E` Ende: `Freud` zählt die beantworteten Fragen zusammen.

BEISPIEL

Man hat alle Fragen beantwortet oder auch einige übersprungen. Nach der letzten Testfrage wird `Freud` durch die Eingabe von „`END`" im Menü beendet. Nun zählt das Programm die jeweiligen Punkte zusammen und gibt diese aus:

```
F1 = 16      F3 = 16      F5 = 16.
```

Die erreichten Punkte müssen nun notiert werden, denn `Freud` speichert sie nicht. Nun muß man das Programm mit [Return] ab-

schließen. Jetzt kann man entweder den obigen Text zur Analyse heranziehen oder das Dokument „Doc.Freud“ lesen. Die Tasten [F1], [F3] und [F5] und deren Punktzahl beziehen sich auf den jeweiligen psychologischen Typus. Viel Spaß bei der Selbstanalyse!

PROBLEME

Beantwortet man nur einzelne Fragen und überspringt einige, dann kann es bei ihrer eventuellen nachträglichen Beantwortung über das `Help`-Menü dazu kommen, daß die Fenster in einer Weise eingeblendet werden, daß der Text nicht lesbar ist.

HERKUNFT

`Freud` befindet sich auf Fishdisk 105. Es ist ein Shareware-Programm. Der Autor beansprucht für sich das Copyright, gestattet aber den nichtkommerziellen Vertrieb. Eine Spende wird gerne entgegengenommen. Seine Anschrift lautet:

```
George Trepal
2650 Alturas Road
Bartow, Florida 33830
USA
```

VERWEISE

Psychoanalytische Literatur

© 1990 technicSupport

FuncKey

Programm zur Belegung der Funktionstasten

AUFRUF

Zuerst muß die Datei `FHandler` am besten in das `C:`-Verzeichnis kopiert werden. Diese Datei wird beim Aufruf von `Funckey` mittels

[run] `FuncKey` [⟨*Dateiname*⟩]

dort gesucht. Starten Sie `FuncKey` von der Workbench durch Anklikken des Icons, kann die Datei `FHandler` auch im selben Verzeichnis wie das Hauptprogramm liegen, wird aber — falls dort nicht vorhanden — ebenfalls im `C:`-Directory gesucht.

OPTIONEN

`run` `FuncKey` wird als Hintergrundprozeß gestartet.

`Dateiname` Bei Angabe eines Dateinamens werden die in dieser Datei enthaltenen Funktionstastenbelegungen gleich beim Start des Programmes eingeladen.

Starten Sie `FuncKey` vom CLI aus ohne einen Dateinamen, erscheint zuerst das Kontrollfeld (siehe Beschreibung). Wird allerdings ein Dateiname angegeben, werden die Tasten sofort entsprechend belegt, und das Kontrollfeld erscheint nicht. Ähnlich verhält es sich bei einem Start von der Workbench durch Anklicken des `FuncKey`-Icons, wobei das Kontrollfeld ebenfalls aktiviert wird. Starten Sie `FuncKey` allerdings durch einen Doppelklick auf das Icon einer `FuncKey`-Datei (in der vorher eingegebene Belegungen gespeichert sind), erscheint das Kontrollfeld nicht automatisch — die Funktionstasten werden automatisch mit den geladenen Makros belegt.

ZWECK

`FuncKey` macht Schluß mit der Sinnlosigkeit der 10 Funktionstasten, die nun in mehreren Kombinationen frei vom Benutzer belegt werden können.

© 1990 technicSupport

BESCHREIBUNG

Die Funktionstasten können entweder mit Texten Ihrer Wahl oder auch mit Spezial-Codes von `FuncKey` belegt werden. Letzere führen dann bestimmte Funktionen von *Intuition* aus, zum Beispiel das Hervorholen eines Fensters oder das Durchschalten einzelner Bildschirme. Zu beachten sind die einzelnen Modi, in die `FuncKey` gerät, wenn Sie es von der Workbench oder vom CLI aus starten. Diese hängen hauptsächlich mit dem *Control Panel* zusammen, das die Bedienungsoberfläche von `FuncKey` repräsentiert.

Dieser Teil des Programmes verbleibt nämlich nur auf Wunsch im Speicher, da für ihn natürlich Platz gebraucht wird. Ist das Kontrollfeld im Speicher vorhanden, kann es mit einem Druck auf [HELP] gleichzeitig mit der rechten [ALT] -Taste zum Vorschein gebracht werden. Ist das Kontrollfeld momentan nicht resident, müssen Sie es durch

```
FuncKey
```

vom CLI oder durch einen Doppelklick auf das `FuncKey`-Icon von der Workbench aus einladen. Erscheinen wird das Kontrollfeld immer auf dem Workbench-Bildschirm, so kann es sein, daß es Ihnen nicht sofort auffällt, wenn Sie sich beispielsweise gerade auf dem Bildschirm einer Textverarbeitung befinden. Normalerweise befindet sich das Kontrollfeld nicht im Speicher, wenn Sie das Programm starten. Rufen Sie `FuncKey` allerdings durch das CLI mittels `run` auf oder laden `FuncKey` durch Anklicken einer entsprechenden Datendatei, ist das Kontrollfeld bis zum ersten Aufruf resident, das heißt im Speicher vorhanden. Nach dem ersten Aufruf müssen Sie das Kontrollfeld aber durch Anwahl von [Resident/Yes] wieder resident machen.

Die Funktionsweise von Funckey

Interessant ist es, hier einmal die Arbeitsweise von `FuncKey` zu betrachten. Die Codes der Funktionstasten werden von `FuncKey` abgefangen und — noch bevor sie an die eigentlichen Intuition-Routinen weitergeleitet werden — durch Ihre neue Belegung ersetzt, so daß

 © 1990 technicSupport

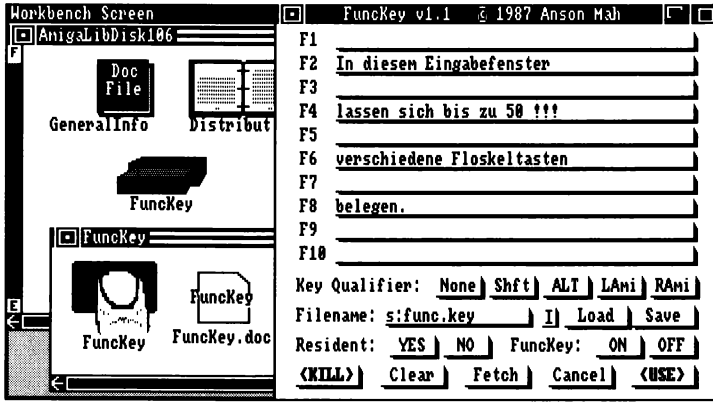
das jeweils verwendete Programm überhaupt nicht „mitbekommt“, daß in Wirklichkeit eine Funktionstaste gedrückt wurde, was sich in der sehr hohen Kompatibilität von FuncKey widerspiegelt. Das heißt im Klartext, daß Sie im CLI ebenso mit den Funktionstasten arbeiten können wie in Ihrer Textverarbeitung, um beispielsweise häufig verwendete Floskeln nicht immer wieder neu tippen zu müssen.

Daß obige Verfahrensweise auch Nachteile haben kann, werden Sie jedoch schon sehr bald bemerken. Benutzt Ihre Textverarbeitung die Funktionstasten nämlich für eigene Zwecke, gibt es Probleme, da die Codes der Funktionstasten von FuncKey nur dann an Intuition weitergeleitet werden, wenn diese im Moment nicht belegt sind. Um dieses Problem elegant zu umgehen, lassen sich alle Belegungen mittels des [FuncKey/On-Off] -Gadgets im Kontrollfeld abschalten, ohne daß diese tatsächlich verloren gehen. Technisch gesehen, werden in diesem Fall die Tastencodes momentan nur nicht durch Ihre spezielle Belegung ersetzt und direkt an Intuition, und damit an Ihre Applikation, weitergegeben.

Die Neubelegung in der Praxis

Die Belegungen für die Tasten werden im Kontrollfeld innerhalb der Zeilen neben „F1“ bis „F10“ eingetragen. Dabei geben die Gadgets am unteren Rand, die sogenannten *Qualifiers* an: die Tasten, die zusätzlich zur angegebenen Funktionstaste gedrückt sein müssen, um den entsprechenden Text oder die Funktion auszugeben beziehungsweise auszulösen. Die Bedeutungen der Abkürzungen dürften wohl eindeutig sein. Die Belegung für eine einzelne Funktionstaste darf übrigens 79 Zeichen nicht überschreiten.

Um die Funktionstasten mit irgendwelchen Sonderfunktionen zu belegen, stehen spezielle Zeichen bereit, die solche Besonderheiten einleiten. So können Sie, indem Sie einem Buchstaben ein „^“ voranstellen, den entsprechenden Control-Code senden, den Code also, den Sie erreichen würden, wenn Sie die Taste mit [CTRL] zusammen drücken würden. Die häufigste Anwendung hierfür wird wohl „^m“ sein, was einem [Return] entspricht, um beispielsweise einen CLI-Befehl abzuschließen.



Das Kontrollfeld von FuncKey

Um mit den Funktionstasten die [Amiga]-Taste zu simulieren, stehen die geschweiften Klammern zur Verfügung. Tragen Sie also bei F1 „{o“ ein, erreichen Sie mit [F1] nun die gleiche Funktion wie mit [linke Amiga] –[O]. Gleiches gilt für „}“ bei der rechten [Amiga]-Taste. Mit „\“ leiten Sie eine der 4 speziellen Intuition-Funktionen von FuncKey ein. So kann mit „\f“ das gerade aktivierte Fenster nach vorne geholt, und mit „\s“ der nächste Bildschirm hervorgeholt und aktiviert werden. Mit „\w“ werden alle momentan auf dem Bildschirm befindlichen Fenster nacheinander aktiviert und — falls „\W“ als Definition für die entsprechende Funktionstaste eingetragen wird, auch ganz nach vorne geholt.

Um eines dieser speziellen Zeichen tatsächlich als Text darstellen zu lassen, muß dieses nur zweimal hintereinander getippt werden. So bekommen Sie bei der Eingabe von „^^“ bei einer Funktionstaste auf dem Bildschirm nur *ein* „^“ zu sehen. Daraus läßt sich folgern, daß die Kombinationen [CTRL]-[^] und [linke Amiga] –[rechte Amiga] mit FuncKey nicht erzeugt werden können.

 © 1990 technicSupport

Bei der Eingabe helfen Ihnen die Felder [Clear], mit dem alle Definitionen vollkommen gelöscht werden, und [Fetch], nach dessen Anklicken die gerade aktuellen Belegungen wieder eingeblendet werden, was bei einem versehentlichen `Clear` von Nutzen ist. Treten bei der Eingabe irgendwelche Fehler auf, wird die Fehlermeldung für ca. eine Sekunde in der Titelzeile des Kontrollfeldes angezeigt.

Mit den [Load]- und [Save]-Gadgets können Sie die derzeitigen Belegungen auf Diskette speichern. Dabei wird der Dateiname verwendet, den Sie neben diesen Feldern unter „Filename" eingetragen haben. Es sollte derselbe sein, den Sie auch beim ersten Aufruf von `FuncKey` angeben. Auf Wunsch kann übrigens (durch Anklicken von [I]) ein Icon für diese Datei erzeugt werden, weshalb die Voreinstellung „`s:Func.Key`" dann nicht sehr sinnvoll ist, da das `s:`-Verzeichnis üblicherweise kein Icon besitzt und `FuncKey` so nicht durch Anklicken der Datendatei eingeladen werden kann. Letzeres funktioniert ohne weiteres auch nur dann, wenn sich das `FuncKey`-Programm im Hauptverzeichnis der Diskette befindet. Dies können Sie auf ein Verzeichnis Ihrer Wahl ändern, wenn Sie das Icon der Datendatei einmal anklicken und dann im `Default Tool`-Feld, das Sie durch Anwahl von `Info` im Workbench-Menü erhalten, den Eintrag von „`:FuncKey`" auf den Pfad Ihrer Wahl ändern. Steht also das Hauptprogramm `FuncKey` der Kürze wegen unter dem Programmnamen `FK` im `System`-Verzeichnis der Boot-Diskette, müssen Sie diesen Eintrag lediglich auf „`SYS:System/FK`" abändern, um `FuncKey` nun auch durch Anklicken der Datendatei laden zu können.

Zum Verlassen des Kontrollfeldes stehen Ihnen mehrere Möglichkeiten offen. Mit [Cancel] werden alle Veränderungen ignoriert, und sämtliche Werte wieder auf den Stand zurück gebracht, den sie hatten, als das Kontrollfeld aufgerufen wurde. [Use] schließt das Kontrollfeld und belegt die Tasten Ihren Wünschen gemäß. Mit [Kill] werden sowohl das Kontrollfeld, als auch der eigentliche *Handler* entfernt, was in einem totalen Löschen des Programmes aus dem Speicher resultiert. Selbstverständlich sind die Funktionstasten nun auch nicht mehr belegt.

PROBLEME

Ein großes Problem entsteht durch die aktuelle Arbeitsweise von `FuncKey`, die oben schon angesprochen wurde. Da die Tastaturcodes nämlich direkt durch `FuncKey` „übersetzt" werden und nicht erst die Intuition-Routinen durchlaufen, wird auch eine mit dem `Setmap`-Befehl eingestellte Tastaturbelegung nicht erkannt. So muß zur Erzeugung des gewünschten Zeichens auf dem Bildschirm immer *der* Buchstabe in der Definition eingetragen werden, der an der Stelle des gewünschten Buchstabens auf der amerikanischen Tastatur stehen würde. Folgende, nicht ganz vollständige Übersicht soll Ihnen dabei behilflich sein und hoffentlich viel Sucharbeit sparen. In der linken Spalte stehen hier die Buchstaben, die in der Definition einzutragen sind, um nach einem „`Setmap d`" auf dem Bildschirm die Buchstaben und Zeichen der rechten Spalte zu erhalten.

z	y	]	+
>	:	{{	Ü
<	;	}}	*
/	-	(	)
&	/	)	=
_	?	*	)
[	ü		

HERKUNFT

Diese Version des Programmes ist auf Fishdisk 106 zu finden. Der Autor

```
Anson Mah
2735 East 2nd Avenue
Vancouver, B.C.
Canada V5M 1E2
```

bittet um einen Beitrag von $5, falls das Programm Gefallen findet. Für insgesamt $10 ist auch die aktuellste Version des Programms und der kommentierte Source-Code zu bekommen.

© 1990 technicSupport

Gagprogramme

Eine Auswahl der besten Programmierspäße

ZWECK

Was wäre das monotone Arbeiten am Bildschirm ohne die vielen kleinen Gags, die das Leben versüßen. Was wäre die Verwaltung eines Girokontos, ohne vorher Zahlen mit dem Mauszeiger abzuschießen und Punkte zu kassieren oder den Bildschirm zu kippen, zu drehen oder zu schmelzen....

Kurzum, die kleinen Gagprogramme sind nicht mehr aus dem Amiga-Alltag wegzudenken. Aus diesem Grund möchte ich Ihnen hier eine Auswahl der gelungensten Programme vorstellen. Der Name in den Klammern ist ein anderer für den gleichen Gag. Die Angabe der Funddiskette ermöglicht das schnelle Suchen des Programms. Ich bin sicher, daß Sie beim Ausprobieren viel Spaß haben werden.

BESCHREIBUNG

Programme für den Mauszeiger:

Diese Gags lassen sich alle per Icon oder vom CLI aus aktivieren und können durch ein Fenster wieder ausgeschaltet werden. Mit Hilfe des Programmes `RunBack` lassen sie sich auch als Background-Task in der Startup-Sequence aufrufen.

Mouse Läßt den Zeiger in allen möglichen Farben scrollen. Vorhanden auf: Amiga Faszination 3/88.

Quickmouse Läßt die Maus bei der kleinsten Berührung über den Bildschirm fliegen. Vorhanden auf: Amiga Faszination 3/88.

Joymouse Mousepointer läßt sich per Joystick steuern. Vorhanden auf: Amiga Faszination 3/88.

Esuom Mauszeiger reagiert spiegelverkehrt zur Eingabe. Vorhanden auf: RPD 53.

© 1990 technicSupport

Target Zeiger verwandelt sich in ein Fadenkreuz, es ertönt ein Schuß beim Drücken der Maustasten. Durch den Feuerknopf wird der Mauspointer wieder normal. Vorhanden auf: Amiga Faszination 1/88, Fish 81, RPD 40.

Ping Mauszeiger hüpft wie ein Ping-Pong-Ball über den Screen. Es ist schwierig bis unmöglich, ihn wieder einzufangen. Vorhanden auf: Panorama 16a.

Introduction Mauspointer wird siebenmal kopiert und zieht die anderen hinter sich her. Vorhanden auf: Amiga Faszination 10/88, RPD 79.

YaBoing II Mauspointer muß Zahlen jagen und fangen. Gibt bei Erfolg Punkte. Vorhanden auf: Fish 136.

Programme für den Screen:

Hier wird der Bildschirm kopiert und vor den anderen gezogen. Dann läuft er — je nach Programm — in sich zusammen oder wird gekippt. Zum Abschalten der Gags muß der (ursprüngliche) Workbench-Screen aktiviert und das Programm ausgeklickt werden, was aber manchmal gar nicht so einfach ist.

Regebogen Läßt einen Regenbogen über den gesamten Screen rollen. Ähnelt dem Marauder-Kopierprogramm. Vorhanden auf: Auge 4000 # 10.

Bröseln Die Fenster beginnen sich aufzulösen. Es schneit auf dem Bildschirm. Vorhanden auf: Auge 4000 # 10.

Schmelzen (Melt)
Der Screen „schmilzt" dahin. Vorhanden auf: Amiga Faszination 1/88, Fish 66, Panorama 16a, RPD 26.

MeltILBM Wie MELT, jedoch nur mit Bildern. Vorhanden auf: RPD 38.

© 1990 technicSupport

Kopf-über (Flip)
Screen wird kopiert und steht auf dem Kopf. Vorhanden auf: Amiga Faszination 1/88, Fish 66, RPD 30.

Blur Der Bildschirm „hüpft". Vorhanden auf: RPD 53.

Tilt Screen wird um 45 Grad gekippt. Vorhanden auf: Fish 54, Amiga Faszination 6+7/88, Panorama 15b, RPD 26.

Flucht (Scat!) Die Windows „fliehen" vor dem Zeiger. Vorhanden auf:
Amiga Faszination 1/88, RPD 53.

150Windows Es werden 150 Fenster eröffnet. Ein sehr langwieriger Prozeß, den man sich vorher gründlich überlegen sollte!!! Vorhanden auf: Amiga Faszination 4+5/88, TBAG 001.

DropShadow Fügt den geöffneten Windows Schatten zu. Der Amiga wird dadurch aber sehr langsam. Vorhanden auf: Fish 87, Amiga Faszination 6+7/88.

Sonstige Gagprogramme:

Unter dieser Rubrik finden sich Programme, die den oben genannten Kriterien nicht untergeordnet werden können. Fast alle können per Icon gestartet werden.

Tastenschloß Tastatur wird solange verriegelt, bis Sie einen richtigen Code (z.B. Amiga) eingeben. Verhindert die Benutzung durch Andere. Vorhanden auf: Auge 4000 # 11.

Mac GAG Läßt Fenster langsam entstehen. Verhindert ruckartiges Auf- oder Abblenden. Vorhanden auf: Auge 4000 # 15.

© 1990 technicSupport

Atari Meets Amiga
Video der Spitzenklasse! Atari 520 ST wird von riesigem Amiga-Ball zerschlagen. Vorhanden auf: Kickstart 40.

ING Vorhandene Windows hüpfen auf dem Screen hin und her. Vorhanden auf: Auge 4000 # 9.

Sand Magnetischer Sand in drei Farben, der vom Mauszeiger angezogen wird. Vorhanden auf: Amiga Faszination 4+5/88, Fish 82.

ViaCom Demonstration von der Qualität des Kabelfernsehens in der Nähe von Los Angeles. Vorhanden auf: Fish 112, Fish 84, Amiga Faszination 4+5/88, RPD 40.

Oing Viele kleine springende Bälle, die über den Screen hüpfen. Vorhanden auf: Fish 33, Amiga Faszination 4+5/88.

Sproing Wie Oing jedoch diesmal mit Klang. Vorhanden auf: Panorama 11a, Fish 33, Amiga Faszination 4+5/88.

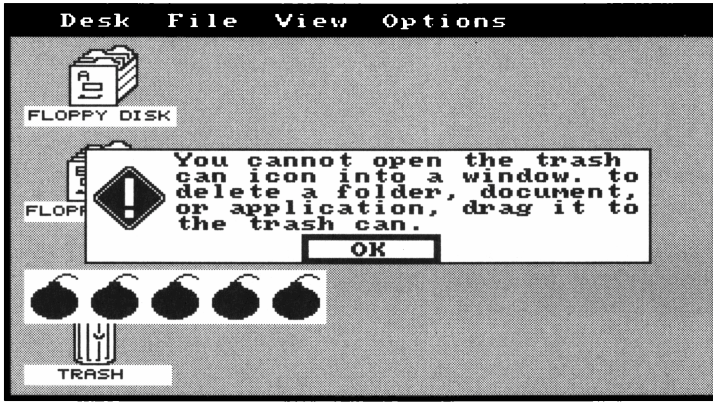
Yaboing Zeigt den Gebrauch von Hardware-Sprites. Vorhanden auf: Amiga Faszination 4+5/88, Fish 36, Panorama 10a.

Zoing Wie Yaboing. Vorhanden auf: Panorama 9a, Amiga Faszination 4+5/88, RPD 2.

Robotroff Läßt ab und zu Roboter auf dem Screen erscheinen, die den Mauszeiger jagen oder „töten“. Vorhanden auf: Fish 59, Amiga Faszination 2/88 + 3/88, RPD 26.

STEmulator Scherzhafte Darstellung eines Atari ST. Vorhanden auf: Fish 43, RPD 13.

© 1990 technicSupport



Auch der Traum vom eigenen Atari kann erfüllt werden

Demolition Ein kleiner Ball zerstört die Fenster. Vorhanden auf: RPD 49.

Muncho Beim Wechseln einer Disk ertönt ein Sound. Vorhanden auf: Amiga Faszination 1/89, Fish 137, RPD 104.

Neeto! Ein böser Scherz für alle CLI-Benutzer: Das Eingabe-Prompt geht spazieren. Vorhanden auf Panorama 027d.

PROBLEME

Manche Gag-Programme lassen sich nicht oder nur sehr schwierig wieder abstellen, so daß bisweilen der Drei-Finger-Griff notwendig wird. Einige der hier aufgeführten Programme vertragen sich nicht miteinander, so daß der Guru in Erscheinung tritt, wenn sie gleichzeitig aktiviert werden.

Hack

Graphisches Abenteuerspiel

AUFRUF

hack kann durch Eingabe von **hack** im **CLI** gestartet werden. Stattdessen können Sie **hack** auch durch Anwählen des **hack**-Piktogrammes von der Workbench aus starten.

Die neue Version 1.0.3D von **hack** liest bei Start Daten von der Datei **hack103.cnf** ein. Diese Datei muß im Verzeichnis **HACK_GAME:** sein!

OPTIONEN

-n Die **news**-Datei wird nicht angezeigt.

-u ⟨*N*⟩ ⟨*N*⟩ wird als Name des Spielers angesehen. Dem Namen direkt folgen kann ein Suffix, nämlich

-T für **Tourist**,

-S für **Speleologist** (Speläologe, Höhlenforscher),

-K für **Knight** (Ritter),

-C für **Cave-man** (Höhlenmensch),

-F für **Fighter** (Kämpfer) oder

-W für **Wizard** (Magier).

Spieloptionen

hack kennt aber auch noch eine weitere Art von Optionen, nämlich solche, die Sie während des Spieles verändern können. Dazu drücken Sie beim Spielen [o] , danach können Sie eine der folgenden Optionen eingeben:

confirm **hack** fragt nach, wenn Sie ein zahmes oder friedliches Monster schlagen, ob dies wirklich Ihre Absicht ist.

silent Die Glocke — beim Amiga ist dies ein Bildschirmblitzen — wird abgeschaltet.

© 1990 technicSupport

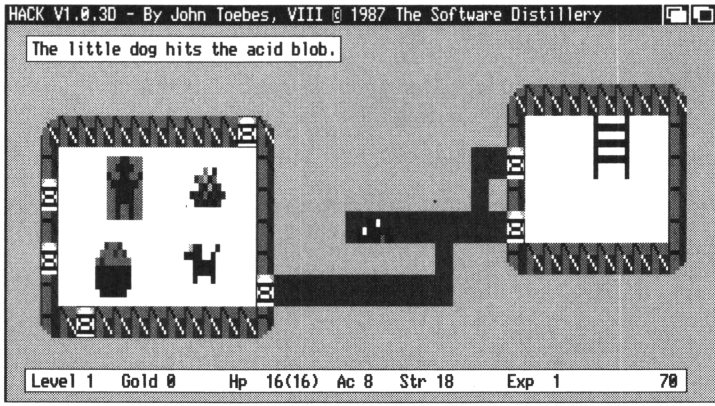
`sortpack`	Ihr Gepäck wird nach Objektgruppen geordnet angezeigt.
`pickup`	Sie nehmen nicht automatisch alles auf, über das Sie laufen, sondern Sie „sehen“ es nur.
`times`	Die abgelaufene „Zeit“ in Spielzügen wird angezeigt.
`restonspace`	Die Leertaste kann verwendet werden, um zu warten.
`male`	Sie teilen `hack` mit, daß Sie männlich sind.
`female`	Sie teilen `hack` mit, daß Sie weiblich sind.
`fixinv`	Einst abgelegte Gegenstände werde im Verzeichnis wieder unter ihrem alten Buchstaben aufgeführt, nachdem sie wiedererlangt wurden.
`tombstone`	Zeigt den Grabstein am Ende des Spieles an.
`packorder`	Hiernach geben Sie die symbolischen Bezeichnungen (siehe unten) für Klassen von Gegenständen ein.

Mit einem vorangestellten Ausrufezeichen `!` verneinen Sie eine Option. So wird mit `!silent` die Glocke eingeschaltet.

ZWECK

Wenn Sie zum ersten Mal `hack` sehen oder selber spielen, werden Sie nicht gleich vor Begeisterung an die Decke springen. Denn `hack` ist ein Spiel, das seine Kostbarkeit nicht sofort für jederman offenbart. Sie müssen schon etwas Geduld und `hack` einige Male gespielt haben, bis Sie so richtig verstehen werden, warum `hack` in der Welt so viele Anhänger hat. Wenn Sie aber erst einmal Ihre Erfahrungen mit `hack` gemacht und dann richtig Feuer gefangen haben, wird `hack` nie langweilig werden. Denn es gibt keine „Lösung“ zu `hack`. Jedes `hack`-Spiel ist anders. Jedesmal erleben Sie wieder neue, einmalige Überraschungen. Während typische Abenteuerspiele höchstens einige hundert mögliche Orte haben, hat `hack` einige zehntausend. In `hack` gibt es hunderte von verschiedenen Objekten — ein wahrer `hack`-Meister kennt nicht nur diese Objekte, sondern auch Ihre Wechselwirkungen.

© 1990 technicSupport



So sieht der Bildschirm bei einem hack-Spiel aus

Diese Beschreibung kann Ihnen nur einige Anregungen zum Spielen von **hack** geben. Ich habe versucht, Ihnen die wichtigsten Grundkenntnisse hier zu vermitteln. Das meiste müssen Sie aber selber herausfinden: Das gehört zum Spiel!

hack ist eine großartige Parabel auf das Leben: Sie beginnen mit wenig Erfahrung und Geld und haben die Möglichkeit, durch richtiges Verhalten immer reicher zu werden. Doch wenn Sie schon denken, jetzt hätten Sie es geschafft, passiert etwas Unvorhergesehenes, etwas Furchtbares, wie eine tote Cockatrice, die plötzlich in Ihrem Weg liegt. Wie sagt doch ein Freund von mir? „Nur die härtesten kommen durch — und davon auch nur die Hälfte.“ Wovon spricht er? Doch nicht etwa vom Weg zum sagenumwobenen **Amulett von Yendor**?

BESCHREIBUNG

Kopieren Sie **hack** zunächst auf eine Diskette, mit der Sie dann arbeiten. Benennen Sie diese Diskette als **HACK_GAME**. Dazu geben

© 1990 technicSupport

Sie im CLI ein:

relabel ? [Return]

Dann legen Sie die **hack**-Diskette in das Laufwerk df0: und tippen:

df0: as "HACK_GAME" [RETURN]

Das ist wichtig, damit Sie wirklich **hack** spielen können. Aber möglicherweise heißt die Diskette bereits **HACK_GAME**, wenn sie bei Ihnen landet. Deswegen können Sie auch versuchen, gleich mit einer Kopie loszuspielen.

Nachdem Sie **hack** mit **hack** vom CLI aus gestartet haben, werden Sie gefragt:

```
Tell me what kind of character you are?
```

Hier können Sie einen Spielcharakter auswählen. Spielen Sie zunächst als Tourist „T“. Wenn Sie später mehr Erfahrung gesammelt haben, probieren Sie andere Charaktere aus!

Sie können den Charakter auch gleich bei Spielstart mit angeben. Durch die Eingabe

```
hack -uStefan-C
```

sagen Sie dem Programm, daß Sie als **Cave-man** „Stefan“ spielen wollen.

Neuere **hack**-Versionen stellen Ihnen diese Frage auch dann nicht, wenn der Spielcharakter bereits in der **hack**-Stardatei **hack103.cnf** angegeben wurde. In diesem Fall legen Sie den Spielcharakter in dieser Datei fest, die Sie mit **ed** bearbeiten können.

Sie können **hack** auch durch Anklicken eines Spielcharakter-Piktogrammes auf dem Arbeitstisch starten. Auch in diesem Fall wird Ihnen die Frage nach dem Charakter dann nicht mehr gestellt.

© 1990 technicSupport

Übersicht der hack-Spielcharaktere		
Charakter	**Stärke**	**Schwäche**
Tourist	reichlich Nahrung und etwas Gold, Wurfpfeile, nützliche Zaubertränke und eine Kamera	wenig Kraft und Schlagpunkte, eine schlechte Schutzklasse und wenig Stärke
Speleologist	Nahrung, gute Werkzeuge, Stärke und Schlagpunkte	wenig Gold, keine Waffen
Knight	sehr gute Rüstung, gute Stärke, nützliche Waffen, ausreichend Schlagpunkte	keinerlei Nahrung, kein Gold, keine Werkzeuge oder Zaubertränke
Cave-Man	sehr gute Schlagpunkte, gute Stärke, akzeptable Waffen und guter Schutz	keine Nahrung, keine Werkzeuge oder Zaubermittel, kein Gold
Fighter	gute Waffen, mittlerer Schutz, Schlagpunkte und Stärke	keine Nahrung, keine Werkzeuge oder Zaubermittel, kein Gold
Wizard	gute Schlagpunkte und Stärke, viele Zaubertränke und Zauberstäbe	keine Nahrung, keine Waffen, keinen guten Schutz

Das Ziel des Spieles

Um `hack` zu „gewinnen", müssen Sie das `Amulett von Yendor` finden, welches irgendwo unter der 20. Ebene verborgen ist. Ich wünsche Ihnen viel Spaß dabei — schicken Sie mir eine Postkarte, wenn Sie es geschafft haben. Es gibt keine Lösung zum „Auswendiglernen", wie bei manchen anderen Spielen. Ich kann Ihnen nicht sagen, wie es geht, weil das Labyrinth und die verschiedenen Monster und Geschäfte jedesmal anders sind. Aber wenn Sie *sehr viel* Erfahrung mit `hack` haben, dann schaffen Sie es vielleicht einmal.

© 1990 technicSupport

Wenn Sie Freunde haben, die auch **hack** spielen, dann kann das eine spannende Herausforderung sein.

Bewegung im hack-Labyrinth

Nach Start des Spieles finden Sie sich im **hack**-Labyrinth wieder. Hier können Sie sich mit den Tasten auf der alphanumerischen Tastatur, bewegen. Legen Sie den Zeigefinger der rechten Hand auf [H], dann ruhen die anderen drei Finger auf [J], [K] und [L]. Sie können sich mit [H] nach links bewegen und mit [L] nach rechts. Mit [J] und [K] geht's nach unten bzw. oben. Auch wenn das gewöhnungsbedürftig ist: Es ist so sehr bequem! Mit [Y], [U], [B] und [N] können Sie sich in den Diagonalenrichtungen bewegen. Dazu müssen Sie nur den Zeigefinger der rechten Hand ein wenig vom [H] entfernen. (Falls Sie einmal mit UNIX arbeiten wollen, so kennen Sie nun auch schon die vi-Editor-Steuerung: Sie funktioniert genauso. Wenn Sie eine deutsche **QWERTZ**-Tastatur benutzen, gibt es leider eine kleine Abweichung durch das Vertauschen von [Y] und [Z].) Sollte Ihnen diese Art der Bewegung nicht gefallen, dann können Sie sich auch mit der Zahlentastatur bewegen. Dabei heißt [5] warten, [8] bewegt Sie nach oben, [2] nach unten, und so weiter.

Drücken Sie statt der Buchstaben [h] [y] [k] [u] [l] [n] [j] [b] einen der großen Buchstaben [H] [Y] [K] [U] [L] [N] [J] [B], das heißt: drücken Sie die [SHIFT]-Taste zusätzlich, so bewegen Sie sich solange in die angegebene Richtung, bis es aus wichtigen Gründen nicht mehr weitergeht. Sie können auch einem der kleinen Buchstaben ein [f] voranstellen — dann geht es auch wiederholt in die angegebene Richtung, aber es wird schon wegen weniger wichtiger Dinge angehalten. Mit einem vorangestellten [m] bewegen Sie sich in die folgende Richtung, ohne automatisch Gegenstände aufzuheben.

hack Programmsteuerung

Die folgenden Befehle sind weniger direkt zum Spielen wichtig als dafür, während des Spieles grundlegende Programmeigenschaften zu kontrollieren.

[Q] *quit* — Abbrechen des Spieles. Sie geben auf.

[S] *save* — Abspeichern des Spieles. Das Spiel wird abgespeichert, um später wieder aufgenommen zu werden. Wenn Sie **hack** später wieder aufrufen, wird das Spiel aus der hier angelegten Datei wieder aufgebaut. Achten Sie auf ausreichenden Disketten-Speicherplatz, denn solche **hack**-Dateien sind oft recht lang.

[o] *options* — Verschiedene Einstellungen können Sie mit [o] ändern. Die wichtigste ist **times**, wenn Sie aktiv ist, wird die Zahl der abgelaufenen Züge angezeigt. **!times** bewirkt das Gegenteil.

[:] Hiermit sehen Sie sich das Feld an, auf dem Sie stehen.

[?] Sie erhalten Informationen über die Bedienung von **hack**.

[/] /⟨*monster*⟩ — Drücken Sie nach dem [/] den Buchstaben eines Monsters, oder klicken Sie mit der Maus ein Monster auf dem Spielfeld an. Sie erhalten den Namen des Monsters, manchmal können Sie danach mit [y] mehr Informationen anfordern.

[!] Hiermit können Sie ein CLI eröffnen.

[v] Version — Die Versionsbezeichnung von **hack** wird angezeigt. Die aktuelle Version ist 1.0.3. (Dann gibt es noch **nethack**, aber das ist eine andere Geschichte . . .)

[CTRL]-[R]
Redraw — Mit [CTRL]-[R] erreichen Sie den Wiederaufbau des **hack**-Bildschirmes.

[CTRL]-[P]
Print — [CTRL]-[P] wiederholt die letzte Nachricht an Sie. Falls Sie dieses öfter drücken, so werden stets noch weiter zurückliegende Nachrichten wiederholt.

Spiel-Kommandos

Nun folgen einige Kommandos, die beim Spielen tatsächlich öfter gebraucht werden. Aber was heißt *Spielen*? Für einige ist **hack** eher

 © 1990 technicSupport

Ernst als Spiel. Ich kenne Leute, die Tage hintereinander **hack** gespielt haben, nur um auf einem System mit mehreren Benutzern einen hohen Punktestand zu haben.

[a] *apply* — Mit diesem Kommando können Sie etwas benutzen, zum Beispiel eine Kamera.

[c] *call* — Hiermit können Sie einen Gegenstand oder eine Klasse von Gegenständen benennen. Falls Sie einen speziellen, einzelnen Gegenstand benennen wollen, dann beantworten Sie die Frage
do you want to name an individual object
mit [y] .

[d] *drop* — Um einen Gegenstand niederzulegen, verwenden Sie dieses Kommando. Mit [d] [2] [b] legen Sie zwei der Gegenstände **b** nieder.

[e] *eat* — Hiermit essen Sie etwas.

[i] *inventory* — Dieses Kommando zeigt Ihnen eine Liste der mitgeführten Gegenstände. Diese Gegenstände sind mit den Buchstaben des Alphabetes „durchnumeriert". Wenn ein anderes Kommando, wie [d] oder [t] , der Angabe eines Gegenstandes bedarf, so müssen Sie den Buchstaben verwenden, der im Verzeichnis vor dem Objekt steht.

[p] *pay* — Wenn Sie in einem Geschäft sind, bezahlen Sie mit [p] ihre Rechnung.

[q] *quaff* — Mit diesem Komando nehmen Sie ein Getränk zu sich.

[r] *read* — Eine Schriftrolle können Sie mit [r] lesen.

[s] *search* — Suchen Sie nach geheimen Türen, Fallen und Monstern.

[t] *throw* — Waffen oder andere Objekte werden mit [t] geworfen. Danach geben Sie eine Richtung mit denselben Tasten an, die Sie auch benutzen, um sich zu bewegen.

[w] *wield* — Um eine Waffe zu führen, benutzen Sie [w] .

© 1990 technicSupport

[z] *zap* — Einen Zauberstab (*wand*) verwenden Sie mit [z].

[C] *christen* — Jedem Tier können Sie einen individuellen Namen geben.

[D] *drop* — Um mehrere Gegenstände möglichst bequem niederzulegen, verwenden Sie dieses Kommando. So führt [D] [a] in der Regel dazu, daß alle Dinge - außer in der Hand getragener Waffen und am Körper angelegter Rüstungen - niedergelegt werden.

[E] *engrave* — Mit [E] werden Nachrichten auf den Boden geschrieben. Mit [E] [-] etwa mit den Fingern.

[P] *put on ring* — Ein Ring wird mit [P] auf den Finger gesteckt.

[R] *remove* — Ein getragener Ring wird so wieder entfernt.

[T] *take off some armor* — Die getragene Rüstung können Sie unter Umständen mit [T] wieder ablegen.

[W] *wear* — Um sich zu bekleiden, verwendet man [W].

[<] *up* — Eine Treppe hochgehen.

[>] *down* — Eine Treppe hinabgehen.

[^] *trap_id* — Bestimmen des Types einer bereits bekannten Falle.

[)] *current weapon* — Die zur Zeit getragene Waffe wird angezeigt.

[[] *current armor* — Es wird angezeigt, welche Rüstungsteile Sie tragen.

[=] *current rings* — Falls Sie Ringe auf den Fingern haben, so erlaubt es dieses Kommando, sie anzuzeigen.

[.] *wait a moment* — Mit dem Punkt [.] oder dem Leerzeichen [ ] warten Sie einen Moment. Um mit dem Leerzeichen warten zu können, müssen Sie allerdings zunächst die Option `restonspace` aktiviert haben.

© 1990 technicSupport

Die Namen von Objektgruppen

An manchen Stellen können Sie Objektgruppen eingeben, etwa um ein Verzeichnis gewisser Objekte zu erhalten oder in der `hack.-103cnf`-Datei die Reihenfolge für die Verzeichnislistung anzugeben. Die Objektklassen werden in `hack` einfach mit Sonderzeichen benannt. Früher wurden diese Zeichen auch verwendet, um Objekte auf dem Bildschirm darzustellen. Eine `hack`-Version, die das noch so macht, finden Sie übrigens auf fishdisk 007.

Hier ist nun eine Auflistung der Zeichen für Objektgruppen:

)	Waffen	?	Schriftrollen
[	Rüstung	(	Werkzeuge
!	Zaubergetränke	/	Zauberstäbe
=	Ringe	%	Nahrung

Berechnung der Punkte

Falls Sie versuchen wollen, möglichst viele Punkte zu erhalten, dann sollten Sie wissen, wie diese errechnet werden. Die wirkliche Berechnung ist recht kompliziert, aber eine grobe Regel ist, daß Ihre Punkte gleich dem Gold sind, welches Sie bei sich tragen, zuzüglich dem Vierfachen Ihrer Erfahrung (eine interne Größe, die das neue `hack 1.0.3D` nicht mehr direkt anzeigt). Diese interne „Erfahrung" steigt jedesmal, wenn Sie ein Monster töten, und zwar um so mehr, je stärker das Monster war. Diese Erfahrung bewirkt auch ab und zu eine Erhöhung der angezeigten, externen Erfahrung, welche wiederum Ihre Treffsicherheit erhöht. Schließlich wird auch noch der Wert der Edelsteine hinzugerechnet, die Sie mit sich führen. 10% der Punkte werden Ihnen abgezogen, wenn Sie gestorben sind. Jedes Amulett, das Sie mit sich führen, bringt Ihnen einen Bonus-Faktor. (Ja! Ich hatte schon einmal zwei Amulette!)

Die hack Start-Datei

In der Datei `hack103.cnf` können Sie verschiedene Standardwerte festsetzen. `hack` schaut nun bei jedem Start in dieser Datei nach, so daß Sie diese Werte nicht jedesmal neu einstellen müssen. Die Datei

© 1990 technicSupport

`hack103.cnf` sollte beim Aufruf von `hack` im aktuellen Verzeichnis sein.

Im folgenden sage ich Ihnen, was in der Start-Datei `hack103.cnf` stehen darf. Sie können diese mit einem Editor (wie `ed` oder `emacs`) erstellen oder ändern. Meistens ist es aber nicht notwendig, *irgendetwas* in dieser Datei zu ändern, da diese bereits mit sinnvollen Voreinstellungen geliefert wird. Eine Ausnahme von diesem Grundsatz ist Ihr Name, den Sie in die Datei `hack103.cnf` einsetzen sollten.

Sie können in `hack103.cnf` die unter „Spieloptionen" aufgeführten Optionen festlegen, indem Sie diese mit einem vorangestellten `OPTIONS=` in eine Zeile schreiben, beispielsweise

```
OPTIONS=!sortpack
```

um das Sortieren Ihres Gepäckes auszuschalten oder

```
OPTIONS=pack:%)!?
```

um festzulegen, daß Ihr Gepäck in der Reihenfolge „Amulett, Essen, Waffen, Getränke, Textrollen" sortiert wird. Besonders wichtig ist die Option `name`, mit der Sie Ihren Namen eingeben können, beispielsweise in der Form

```
OPTIONS=name:DELTOR-F
```

wenn Sie unter dem Namen `DELTOR` spielen wollen. Durch das `-F` ist gleichzeitig festgelegt, daß `DELTOR` als *Fighter* spielt.Mit dem Befehl

`HACKDIR=`⟨*Verzeichnis*⟩

informieren Sie `hack` über das Verzeichnis der Unterstützungsdateien `help`, `hh`, `data`, `record` und `rumors`. Hier ist dies beispielsweise das Verzeichnis `c:/games/hack`.

`LEVELS=`⟨*Verzeichnis*⟩

bewirkt, daß die einzelnen Aufzeichnungen der Höhlenebenen in das angegebene ⟨*Verzeichnis*⟩ gelegt werden sollen.

Sie können mit der Zeile

`SAVE=`⟨*Verzeichnis*⟩

© 1990 technicSupport

das Verzeichnis spezifizieren, in dem Ihr Spiel aufgehoben werden soll, wenn Sie es mit [S] unterbrochen haben.

Diese Beschreibung läßt bewußt vieles offen, denn die Freude beim Spielen von **hack** ist es gerade, die Bedeutung und Funktion all der Monster und Objekte selber festzustellen. Ich wünsche Ihnen viel Vergnügen dabei!

DATEIEN

Folgende Dateien spielen eine wichtige Rolle für die Funktion des Programmes:

`hack`	Das Programm selber
`data`	Eine Datendatei für `hack`
`help`	Eine Datei mit Hilfsinformationen
`rumors`	Eine Datei mit Texten für Glückskeckse
`record`	Die Liste der Punkte der einzelnen Spieler
`Saved Games`	Ein Verzeichnis für unterbrochenen Spiele
`bones_dd`	Beschreibung der Geister

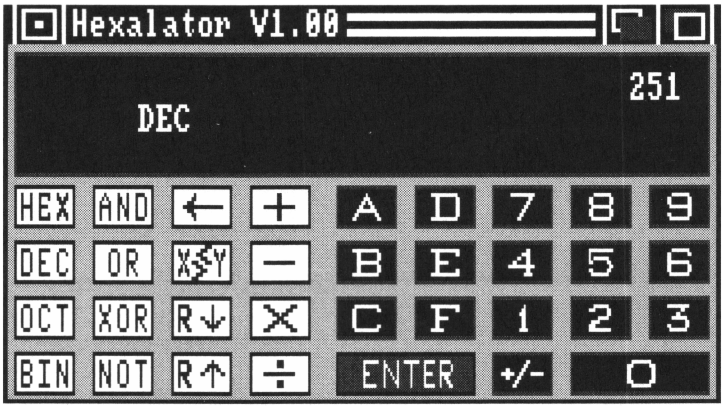
HERKUNFT

Die besprochene Version des Spieles findet sich auf der Fishdisk 062. Das Original stammt von: JAY FENLASON, KENNY WOODLAND, MIKE THOME und JON PAYNE. Es wurde weiterentwickelt von ANDRIES BROUWER. JOHN TOEBES erstellte die Amiga-Version. Die Anschrift von John A. Toebes lautet:

```
John A. Toebes, VIII
120 H Northington Place
Cary
NC 27511
USA
```

VERWEISE

`HackLite` (Fish 187), `NetHack` (Fish 189/190), `Larn` (Fish 63), `Moria` (RPD 132)



Wieviel ist $F0 + %1011?

Hexalator

Umrechnen verschiedener Zahlenformate

AUFRUF

`Hexalator` kann vom CLI aus durch Eingabe von

[run] `Hex`

oder von der Workbench durch Anklicken des Icons gestartet werden.

ZWECK

`Hexalator` dient dem Rechnen innerhalb verschiedener und zwischen verschiedenen Zahlensystemen und ist daher besonders für Programmierer interessant.

 © 1990 technicSupport

BESCHREIBUNG

Nach Aufruf von `Hexalator` erscheint ein Fenster, in dem ein Taschenrechner mit Display und Tastatur dargestellt ist. Die Bedienung dieser Tastatur erfolgt auf einfache Weise mit dem Mauszeiger. Hexalator erlaubt die Eingabe von Zahlenwerten in dualer, oktaler, dezimaler und hexadezimaler Form sowie das Umrechnen dieser Zahlen in ein anderes Format. Zusätzlich zu den Grundrechenarten stehen die logischen Operatoren AND, OR, EXOR und NOT zur Verfügung.

BEISPIEL

Sie möchten wissen, welche Dezimalzahl sich als Summe aus %1011 und $F0 ergibt? Betätigen Sie nacheinander die Tasten [HEX], [F], [0], [BIN], [ENTER], [1], [0], [1], [1], [+], [DEC] und Sie erhalten das Ergebnis: `251`.

HERKUNFT

Das Programm befindet sich auf der RPD-Disk 66 und auf Fishdisk 067. Autor ist SCOTT S. ROSS.

Hyperbase

Menügesteuerte Dateiverwaltung

AUFRUF

hyperbase [⟨*Dateiname*⟩] [⟨*Zahl*⟩]

OPTIONEN

Dateiname — Der Name der Datei, in der alle Datensätze gespeichert sind. Sie muß sich unter dem Namen ⟨*Dateiname*⟩.dat im aktuellen Verzeichnis befinden.

Zahl — Diese Zahl legt die Anzahl der Datensätze fest, die im Puffer gehalten werden. Die Puffergröße wird entsprechend eingerichtet.

ZWECK

Hyperbase ist ein Dateiverwaltungsprogramm, das weitgehend von der Workbench aus zu bedienen ist und für einfache Aufgaben der Dateiverwaltung durchaus geeignet ist. Lediglich die Erstellung von Masken und Formularen muß mit Hilfe eines Editors erfolgen. Wegen der Übersichtlichkeit ist **Hyperbase** auch ein ausgezeichnetes Lernobjekt für die grundsätzlichen Arbeitsprinzipien dieser Softwaregattung.

BESCHREIBUNG

Zu den grundsätzlichen Aufgaben einer Dateiverwaltung gehört es, Datensätze zu speichern, einzelne Datensätze oder Gruppen von Datensätzen nach bestimmten Kriterien zu selektieren sowie diese Datensätze in einer frei bestimmbaren Form zu Papier zu bringen. Eine solche *logische* Datei besteht daher gewöhnlich aus mehreren *pysikalischen* Einzeldateien, die wir im folgenden zur besseren Unterscheidung mit dem englischen Begriff *File* bezeichnen werden. Eine **Hyperbase**-Datei besteht aus mindestens zwei Files:

© 1990 technicSupport

⟨*Dateiname*⟩.fmt Dieses File bestimmt den Aufbau der Datei. In ihm sind alle notwendigen Informationen über Anzahl, Größe und Art der Datenfelder sowie über die Gestalt der Eingabemaske untergebracht.

⟨*Dateiname*⟩.dat Dieses File mit der Endung „.dat“ enthält die gespeicherten Datensätze, das heißt die eigentliche Information.

Es ist aufschlußreich, ein solches „.dat“-File mit dem `type`-Befehl auszugeben oder in einen Editor zu laden, um einmal zu sehen, wie `Hyperbase` seine Daten verwaltet:

```
H"Dr.","Klausmann","Klaus","D","4234","Clausthal",
                                "Harzer Weg 425a"...,,
H"Herr","Petersen","Peter","D","2000","Hamburg",
                                "Am Peterswall 345a",...
```

Wie man sieht, entspricht jede Zeile genau einem Datensatz. Die Inhalte der einzelnen Datenfelder sind von Anführungszeichen eingeschlossen, einzelne Datenfelder sind durch Kommata voneinander getrennt. Jeder Datensatz beginnt mit einem „H“, welches den Datensatz als gültig, oder einem „D“, welches ihn als gelöscht markiert. An der unterschiedlichen Länge der Einträge erkennt man außerdem, daß `Hyperbase` ein Datenfeld nicht mit Leerzeichen auffüllt, was im Einzelfall sehr viel Platz sparen kann.

Zwei weitere File-Typen sind ebenfalls von Bedeutung:

⟨*Dateiname*⟩.prt Ein Formular, welches die Form festlegt, in der die Datensätze beim Ausdruck zu Papier gebracht werden.

⟨*Dateiname*⟩.ind Dieses File enthält die Indexdatei, die aus dem Programm heraus mit der `Sort`-Funktion erzeugt werden kann. Sie bestimmt die Aufeinanderfolge der Datensätze.

`Hyperbase` gestattet es, für ein und dieselbe Datei mehrere „.fmt“- und „.prt“-Files anzulegen und wahlweise zu laden. So kann für eine Adreßdatei ein Formular zum Ausdrucken von Etiketten, ein

anderes aber für einen Serienbrief erstellt werden. Dabei muß jedoch peinlichst genau darauf geachtet werden, daß die verschiedenen Masken und Formulare hinsichtlich Anzahl und Größe der Datenfelder übereinstimmen. Das Öffnen einer Eingabemaske, die beispielsweise über mehr Eingabefelder verfügt als die vorherige, kann die gesamte Dateistruktur hoffnungslos durcheinanderbringen.

Anlegen einer Adreßdatei

Das Arbeiten mit `Hyperbase` kann an einem Beispiel besser vermittelt werden als durch eine trockene Auflistung aller Befehle, weshalb wir Schritt für Schritt eine Adreßdatei erstellen werden.

Es empfiehlt sich, das `Hyperbase`-Programm von der Fishdisk auf eine neu formatierte Diskette zu kopieren, um genügend Platz für die Daten zu haben. Für kleinere Dateien mag es praktischer sein, eine Kopie der Workbench-Diskette anzufertigen, auf dieser alle nicht benötigten Files zu löschen und ein Verzeichnis für Hyperbase anzulegen.

Es ist am zweckmäßigsten, alle Files in dem Verzeichnis abzulegen, in dem sich auch das `Hyperbase`-Programm befindet. Daher sollte man vor den folgenden Arbeitsschritten mit dem Befehl

`cd` ⟨*Verzeichnis, in dem sich Hyperbase befindet*⟩

in das `Hyperbase`-Verzeichnis wechseln.

Unsere Adreßdatei soll folgende Daten erfassen: Anrede (10 Zeichen), Name (30 Zeichen — Doppelnamen sind ja zur Zeit sehr beliebt!), Vorname (15), Staat (3), PLZ (5), Ort(25), Straße (30), Telefon (15, numerisch), Geburtsdatum (Format: xx.xx.xxxx). Diese Informationen sowie die Position der Eingabefelder auf dem Bildschirm legen wir in der Datei „Adressen.fmt“ ab, wozu wir den Systemeditor `Ed` verwenden können:

```
ed Adressen.fmt
```


© 1990 technicSupport

Das „.fmt“-File

Der Aufbau eines solchen „fmt.“-Files muß sich strengstens an die Konventionen halten, die durch das `Hyperbase`-Programm vorgegeben werden. Jeder Eintrag muß aus genau sieben Elementen bestehen, die durch Kommata voneinander zu trennen sind. Aufgrund dieser Trennfunktion läßt sich das Kommazeichen nur darstellen, indem ihm ein Backslash (\) vorangestellt wird. Dasselbe gilt für Anführungszeichen und den Backslash selbst. Werden einzelne Elemente weggelassen, muß das zugehörige Komma in jedem Falle gesetzt werden. Hier die sieben Elemente:

1) Name des Datenfeldes

2) Zeilenposition

3) Spaltenposition

4) Bei dieser Programmversion immer „1“

5) Ausgabeanweisung für 6)

6) Formel für Ausgabe

7) Eingabeanweisung

Da wir eine Eingabemaske erstellen wollen, sind zunächst nur die Punkte 1, 2, 3 und 7 von Interesse. Der Name des Datenfeldes (1) erscheint auf dem Bildschirm an der Position, die durch 2 und 3 festgelegt wird. Hinter diesem Namen erscheint das Eingabefeld, das durch eine der Datenfeldlänge entsprechende Anzahl von „_“-Zeichen (für alphanumerische Eingabe) bzw. „#“-Zeichen (ausschließlich für Ziffern) definiert ist. Jedes Eingabefeld sollte mit einem „^“ abgeschlossen werden, da in diesem Fall bei der Datenerfassung ein [Return] erforderlich wird, um in das nächste Feld zu gelangen. Andernfalls kann es leicht geschehen, daß man über das Ende eines Feldes hinaus in das nächste Feld hineinschreibt. Die

© 1990 technicSupport

Maske für unsere Adreßdatei kann dann folgendermaßen aussehen:

```
Anrede: ,3,3,1,,,__________^,
Name: ,5,3,1,,,_______________________________^,
Vorname: ,7,3,1,,,_______________^,
Staat: ,11,3,1,,,___^,
PLZ: ,11,16,1,,,_____^,
Ort: ,11,30,1,,,_________________________^,
Straße: ,13,3,1,,,_______________________________^,
Tel: ,16,3,1,,,################^,
Geburtsdatum: ,18,3,1,,,##.##.####^,
```

Zu beachten ist, daß auch die letzte Eingabezeile im Editor mit einem Return abgeschlossen werden muß, da `Hyperbase` sonst eine Fehlermeldung ausgibt. Nach dem Abspeichern der Maske kann Hyperbase mit dem Befehl

```
hyperbase Adressen
```

aufgerufen werden.

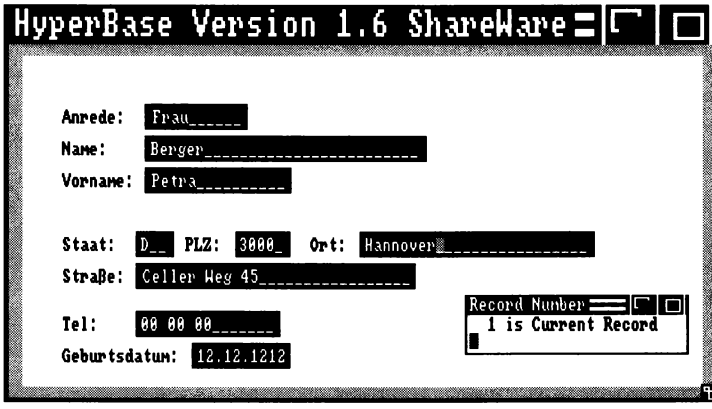
Dateneingabe

Nach diesem Aufruf öffnet Hyperbase ein Titelbild, welches über die momentan geladenen Files für Datei, Index, Maske und Formular informiert, ebenso über Anzahl der Datensätze, Nummer des aktuellen Datensatzes, Zahl der gefundenen Datensätze und die Größe des Zwischenspeichers. Alle Programmfunktionen stehen jetzt in der Menüleiste bereit, die ihre Gestalt — abhängig vom jeweiligen Arbeitsmodus — ändert. In den Menüs sind die Tastaturkombinationen aufgeführt, die als Alternative zur Maussteuerung benutzt werden können.

Zunächst wählen wir die Funktion `Insert` aus dem `Browse`-Menü, die für die Eingabe neuer Datensätze zuständig ist. Die Eingabemaske erscheint, und die Menüleiste stellt unter den Punkten `File`, `Browse` und `Edit` folgende Funktionen zur Verfügung:

`Redraw` Baut den Bildschirm neu auf.

 © 1990 technicSupport



So sieht die Eingabemaske auf dem Bildschirm aus

`Save`	Kehrt unter Abspeichern der neuen Eingaben in das Hauptmenü zurück.
`Quit`	Kehrt ohne Abspeichern in das Hauptmenü zurück.
`Forward`	Springt unter Abspeichern des aktuellen Datensatzes zum nächsten.
`Back`	Springt einen Datensatz zurück.
`Goto`	Springt zu einem Datensatz mit einer bestimmten Nummer.
`Kill Field`	Löscht das Eingabefeld, in dem sich der Cursor befindet.
`Delete`	Markiert einen Datensatz als gelöscht.
`Undelete`	Ist im Insert- und im View-Modus inaktiv.

Im Insert-Modus kann man sich mit der Maus, mit Hilfe der Cursortasten sowie mit [Return] und [Backspace] durch die Maske bewegen

und seine Daten eingeben. Ist ein Datensatz komplett ausgefüllt, gelangt man mit `Forward` zum nächsten Datensatz. Solange keine Sortierung vorgenommen wurde, wird bei der Eingabe immer die Datensatznummer 1 angegeben, was anfangs etwas irritieren kann. Nachdem alle Datensätze eingegeben sind, kehrt man mit `Save` in das Hauptmenü zurück.

Sehr ähnlich ist die `View`-Funktion, die zum Durchblättern und Sichten einer Datei dient. In diesem Modus lassen sich Datensätze ändern. Nach jeder Änderung eines Datensatzes ist dann `Save` aufzurufen, da beim Weiter- oder Zurückblättern die Änderungen verloren gehen.

Sortierfunktion und Indexdatei

Im `Sort`-Menü läßt sich mit der gleichnamigen Option die Reihenfolge der Datensätze festlegen. Es erscheint ein Prompt, und es können die Nummern der Datensätze eingegeben werden, nach denen sortiert werden soll. Die Nummern ergeben sich aus den Positionen der Datenfelder im „.fmt"-File. Mit einem Punkt wird das Ende der Eingabe gekennzeichnet. Soll unsere Adreßdatei nach dem Namen sortiert werden und bei Namensgleichheit der Vorname entscheiden, lautet die Eingabe:

2 [Return]
3 [Return]
. [Return]

Ein so erstelltes Indexfile läßt sich mit der `Save Index`-Funktion unter dem Namen „Adressen.ind" abspeichern. Die Sortiergeschwindigkeit wird übrigens in entscheidendem Maße von der Größe des Cache-Speichers bestimmt, der für 20 Datensätze voreingestellt ist, beim Aufruf aber entsprechend vergrößert werden kann.

Suchfunktionen

Die Qualität der Suchfunktion ist das entscheidende Kriterium für den Nutzen einer Datenbank. Auch wenn `Hyperbase` keine so komplexen Selektierungen vornehmen kann wie etwa Superbase, dürften

© 1990 technicSupport

die Suchmöglichkeiten dieses Programmes für die meisten Anwendungen ausreichen.

Nach dem Aufruf der `Search`-Funktion aus dem gleichnamigen Menü erscheint eine leere Eingabemaske, die einen wesentlichen Unterschied zu der Maske des Insert- und View-Modus aufweist. Aus jedem Eingabefeld kann man sich mit Hilfe der linken Cursortaste um genau eine Stelle nach links herausbewegen. An dieser Stelle werden nur vier Zeichen akzeptiert, die in reverser Schrift dargestellt werden, nämlich die folgenden Vergleichsoperatoren:

=	Gleich
!	Ungleich
>	Größer oder gleich
<	Kleiner oder gleich

Nach der Eingabe eines dieser Zeichen befindet man sich wieder auf der ersten Stelle des jeweiligen Eingabefeldes und kann den Buchstaben oder den Begriff eintragen, mit dem verglichen werden soll. Werden solche Einträge in mehreren Feldern vorgenommen, so werden nur die Datensätze herausgesucht, die alle diese Kriterien erfüllen. Um alle Berliner zu selektieren, bringt man den Cursor also in das Feld „PLZ“ und betätigt nacheinander die Tasten [←], [=], [1], [0], [0] und [0]. Verläßt man die Maske mit der `Quit`-Funktion, zeigt `Hyperbase` im Titelbild in der Zeile `Last Search found` die Anzahl der gefundenen Datensätze, die jetzt am Bildschirm betrachtet (`Display Screen`), ausgedruckt (`Display Printer`) oder gelöscht (`Delete Found`) werden können. Ebenso ist es möglich, wenn auch sehr gefährlich, alle nicht gefundenen Datensätze zu löschen (`Delete not Found`). Da vor dem Löschen keine Sicherheitsabfrage erfolgt, sollte man bei der Führung des Mauszeigers besonders vorsichtig sein: Nur allzu leicht rutscht man von der Display-Funktion in die darunterstehende Löschoption.

Sollte einmal ein solches Mißgeschick passieren, dann muß man nicht gleich zur HB greifen, denn die gelöschten Datensätze sind nicht völlig verloren: Sie sind im „dat.“-File lediglich mit einem vorangestellten „D“ als gelöscht markiert. Man kann sie daher mit der

`Found deleted rec`-Funktion wiederfinden und mit `Display Screen` betrachten. Jetzt kann auch die `Undelete`-Funktion aktiviert werden, mit der sich die Datensätze zurückholen lassen. Nach jedem Restaurieren einer Datei sollte das „.ind"-File mit Hilfe der `Sort`-Funktion neu erstellt werden, andernfalls kann `Hyperbase` nicht alle Datensätze finden.

Das File-Menü

Das `File`-Menü stellt alle erforderlichen Ein- und Ausgabeoperationen zur Verfügung:

`Open` Öffnet eine neue Datei mit allen vier dazugehörigen Files, sofern diese vorhanden sind. Anzugeben ist der reine Dateiname ohne die Endung „.dat".

`Open file` Ermöglicht in einem Untermenü das separate Laden eines „.dat"-, „.fmt", „.prt" oder „.ind"-Files. Auch hier brauchen die Endungen nicht angegeben zu werden.

`Print` Alle Datensätze werden über das gültige „.fmt"-File ausgedruckt. Der Druckvorgang beginnt gewöhnlich erst, nachdem man `Hyperbase` mit `Quit` verlassen hat.

`New` In dieser Programmversion noch ohne Funktion.

`Cleanup` Dient dazu, als gelöscht markierte Datensätze endgültig zu löschen, das heißt, sie auch physikalisch zu entfernen. Bei mir endeten diese Versuche jedoch bisweilen mit einem Guru und der netten Meldung, daß meine Festplatte „not validated" sei. Es ist daher sicherer, das „.dat"-File von Zeit zu Zeit in einen Editor zu laden und dort alle Zeilen zu löschen, die mit einem „D" beginnen. Hier lassen sich auch andere Änderungen vornehmen, allerdings müssen die Kommata und Anführungszeichen unangetastet bleiben, da sie die interne Struktur der Datei bestimmen. Die leichte Editierbarkeit des „.dat"-Files erweist sich hier als besondere Stärke von Hyperbase.

© 1990 technicSupport

Die Erstellung von Formularen

Zum Abschluß erstellen wir noch ein Formular für die Druckausgabe, welches die in der Datei gespeicherten Anschriften in Form von Etiketten ausgibt. Das Verfahren entspricht der Herstellung einer Maske, nur spielen jetzt die Elemente 5 und 6 die entscheidende Rolle. Die Ausgabeanweisung (5) legt Anzahl und Art der auszugebenden Zeichen fest, während der Inhalt der Ausgabe durch eine Formel (6) bestimmt wird. Der Inhalt eines Datenfeldes wird dabei durch den Ausdruck „{x}“ referiert, wobei x die Nummer des Datenfeldes bezeichnet. Mit dem „&“-Zeichen lassen sich Strings verknüpfen, mit den eckigen Klammern Teilstrings bilden. Laut Manual sollte die Operation

```
({1} & {2}) [3,6]
```

das Ergebnis „cdAB“ liefern, wenn Feld 1 den Ausdruck „abcd“ und Feld 2 „ABCDEFG“ enthält. Entgegen dieser Anleitung stößt sich **Hyperbase** jedoch an dem Komma und kann diesen Audruck nicht auswerten. Als Ausweg kann man einzelne Charakter extrahieren und diese dann aneinanderfügen:

```
{1}[3]&{1}[4]&{2}[1]...
```

Für arithmetische Verknüpfungen stehen die Operatoren „+“, „-“, „*“ und „/“ zur Verfügung. Ein erster Entwurf für den Etikettendruck kann daher folgende Gestalt haben:

```
,2,1,1,__________,{1},,
,3,1,1,____________________________________________,{3}&" "&{2},
,4,1,1,,,,
,5,1,1,________________________________,{7},,
,6,1,1,____________________________,{4}&"-"&{5}&" "&{6},,
,7,1,1,,,,
,8,1,1,,,,
```

Unter dem Namen „Adressen.prt“ abgespeichert, liefert dieses For-

mular das Ergebnis:

```
Anrede
Vor- und Nachname
<Leerzeile>
Straße
Staat, PLZ und Ort
<Leerzeile>
<Leerzeile>
```

Die Anzahl der Leerzeilen muß natürlich, entsprechend dem Abstand der verwendeten Aufkleber, variiert werden. Komplizierter wird es, wenn man den Zeilenabstand variieren, Unterstreichungen vornehmen oder andere Parameter ändern möchte. Im Manual findet sich der lapidare Hinweis, daß man lediglich „cntl-L“ oder ähnliche Steuerbefehle als Element 1 einfügen bräuchte. Zunächst wunderte ich mich darüber, daß kaum eines der Steuerzeichen, die ich eingefügt hatte, eine Wirkung zeigte. Ich hatte — vielleicht eine Nachwirkung der seligen 64er-Zeit — die druckereigenen Steuersequenzen benutzt, ohne zu berücksichtigen, daß die Ausgabe ja über das `printer.device` läuft, das bekannterweise eigene Steuercodes benötigt, die wiederum weniger bekannt sind. Aus diesem Grunde schließt sich an diese Beschreibung eine Übersicht über die wichtigsten Steuerzeichen für das `prt:`-Gerät an. Unter diesen Steuerzeichen ist jedoch auch der ESC-Code zu finden, der dem ASCII-Wert 27 entspricht und mit vielen Editoren nicht eingegeben werden kann. In diesem Buch wird daher der Editor `Dme` besprochen, der über diese Möglichkeit verfügt. So können wir unser Formular zum Beispiel dahingehend abändern, daß die letzte Zeile unterstrichen ausgegeben wird:

```
,6,1,1,_____________________________,"<Esc>[4m"&{4}&"-"&{5}&"
                                    "&{6}&"<Esc>[24m",,
```

`<Esc>` steht hier für den Wert 27. Auf diese Art und Weise lassen sich die verschiedensten Funktionen aufrufen. Voraussetzung ist dabei, daß in den Preferences der richtige Druckertyp eingestellt ist, und der angeschlossene Drucker über die entsprechende Funktion verfügt.

© 1990 technicSupport

Steuersequenzen für das printer.device

Reset	Escc	Zeile vorwärts	EscD
Return, Zeile vorwärts	EscE	Zeile rückwärts	EscM
Kursiv ein	Esc[3m	Kursiv aus	Esc[23m
Unterstreichen ein	Esc[4m	Unterstreichen aus	Esc[24m
Fettschrift ein	Esc[1m	Fettschrift aus	Esc[22m
Elite ein	Esc[2w	Elite aus	Esc[1w
Condensed ein	Esc[4w	Condensed aus	Esc[3w
Enlarged ein	Esc[6w	Enlarged aus	Esc[5w
Doppeldruck ein	Esc[4"z	Doppeldruck aus	Esc[3"z
Hochstellen ein	Esc[2u	Hochstellen aus	Esc[1u
Tiefstellen ein	Esc[4u	Tiefstellen aus	Esc[3u
Deutscher Zeichensatz	Esc(K	NLQ ein	Esc[2"z
NLQ aus	Esc[1"z		

PROBLEME

Der große Schwachpunkt des Programms zeigt sich bei der Dateneingabe: Wie so viele aus den Staaten kommende Programme akzeptiert auch Hyperbase keine deutschen Umlaute. Wie man dieses Problem einigermaßen elegant umgehen kann, läßt sich in der PD-Story nachlesen. Abgesehen von der `Cleanup`-Funktion zeigte sich `Hyperbase` absturzsicher. Dennoch sollte man regelmäßig Sicherheitskopien von der „.dat"-Datei anlegen.

HERKUNFT

Die hier beschriebene Version 1.6 des Programms findet man auf der Fishdisk 131. `Hyperbase` ist Shareware. Für $25.00 wird man registrierter Benutzer und erhält von den Autoren MICHAEL MACKENZIE, MARC MENGEL und CRAIG NORBORG den neuesten C-Quellcode (Manx 3.4a). Die Anschrift:

```
General Design
P.O. Box 2352, West Lafayette,  IN  47906, USA
```

IconType

Ändert den Typ eines Icons

AUFRUF

`Icon Type` wird aus dem CLI mit dem Befehl

`IconType t|p|r|d|g` ⟨*Iconname*⟩

aufgerufen.

OPTIONEN

`t`	Erzeugt ein TOOL-Icon
`p`	Erzeugt ein PROJECT-Icon
`r`	Erzeugt ein DRAWER-Icon
`d`	Erzeugt ein DISK-Icon
`g`	Erzeugt ein GARBAGE-Icon
`Iconname`	Name der `.info`-Datei, die die Iconstruktur enthält. Das Suffix „.info“ ist nicht mitanzugeben.

BESCHREIBUNG

Icon ist nicht gleich Icon. Die Workbench benötigt für unterschiedliche Aufgaben auch verschiedene Icontypen: `Disk`-Icons, die das Fenster für das Basisverzeichnis einer Diskette öffnen; `Drawer`-Icons, die Fenster für Unterverzeichnisse öffnen; `Tool`-Icons zum Starten von Programmen sowie `Project`-Icons, die Daten gemeinsam mit dem zugehörigen Programm laden. Als „Prototypen“ lassen sich nennen: das Piktogramm einer Fishdisk, das der System-Schublade, das Preferences-Icon und das Icon eines mit Notepad erstellten Textes. Nur selten dürften wohl Icons vom Typ `Garbage` benötigt werden, die für „Mülltonnen“-Verzeichnisse zuständig sind.

Mit `IconType` lassen sich beliebige `.info`-Dateien unter Beibehaltung der Bildstruktur in andere Icontypen umwandeln. Die Verwendung von `IconType` empfiehlt sich insbesondere bei animierten

 © 1990 technicSupport

Icons oder Icons mit besonders großen Piktogrammen, da diese nicht mit dem Workbench-Programm `IconEd` bearbeitet werden können. Der Erfolg der Umwandlung läßt sich mit dem `Info`-Tool aus dem Workbench-Menü überprüfen. Dazu muß das Fenster, in dem sich das geänderte Icon befindet, in jedem Fall neu geöffnet werden, damit `Info` auf die neue Iconstruktur zugreifen kann. Anschließend ist das Icon zu aktivieren und `Info` aufzurufen. Es erscheint ein Fenster, in dem sich unter anderem der Typ des Icons ablesen läßt. Ein Hinweis noch zu den `Disk`-Icons: Diese werden erst sichtbar, nachdem sie in „.info" umbenannt und in das Basisverzeichnis der entsprechenden Diskette kopiert worden sind. Bevor Sie diese Arbeitsschritte wagen, sollten Sie ganz sicher sein, daß es sich wirklich um ein `Disk`-Icon handelt, da ein falscher Icontyp die Diskette für die Workbench unsichtbar macht.

PROBLEME

Das `Info`-Tool läßt sich nur dann aufrufen, wenn zu der `.info`-Datei auch die entsprechende Datei ohne dieses Suffix existiert.

HERKUNFT

Das hier besprochene Programm befindet sich auf Fishdisk 102 (im Unterverzeichnis `Xicon/Icon_Stuff`). Autor ist

```
Pete Goodeve
3012 Deakin Street #D
Berkeley, Calif. 94705
USA
```

VERWEISE

`ZapIcon` (Panorama 11), `IconImage` (Fishdisk 120), `IconAssembler` (Fishdisk 101), `SetAlternate` (Fishdisk 012), `Xicon` (Fishdisk 157), `CLIRun` (RPD 056)

KeyClick

Akustisches Feedback bei Tastatureingaben

AUFRUF

`KeyClick` läßt sich durch Anklicken seines Icons starten. Aus dem CLI heraus wird es durch Eingabe von

```
run[back] KeyClick
```

gestartet.

ZWECK

`KeyClick` läßt beim Betätigen einer Taste oder eines Mausbuttons ein kurzes akustisches Signal ertönen.

BESCHREIBUNG

Während man beim Arbeiten mit einer Schreibmaschine deutlich hört, ob ein Anschlag erfolgt ist oder nicht, kann man sich beim Schreiben auf einer Computertastatur dessen nie ganz sicher sein. `KeyClick` sorgt für Abhilfe, indem es jeden korrekten Anschlag mit einem kurzen Signalton quittiert. Nach dem Starten des Programmes erscheint in der linken oberen Ecke des Bildschirms ein kleines Fenster, in dem mit einem Schieberegler die Lautstärke des Tones geregelt werden kann. Auch ein Ein/-Ausschalter ist vorhanden, der nur mit der linken Maustaste angeklickt werden muß. Über das [•] -Feld wird das Programm wieder verlassen. Möchte man ständig mit dem Programm arbeiten, so muß es zunächst installiert werden. Dieser Vorgang verläuft ganz analog zur Installation von `ExplodingWindows`, nur befindet sich das Programm statt in der `Graphics`- in der `Utilities`-Schublade. Lesen Sie bitte in der Beschreibung zu `ExplodingWindows` nach.

PROBLEME

Wenn das Programm aktiv ist, kann es beim Aufrufen der `Preferences` zu einem Systemabsturz kommen.

© 1990 technicSupport

HERKUNFT

Beschrieben wurde die Version 1.1 des Programmes, die man auf der TBAG 26 findet. Sie stammt von `Mike Duppong`, der über das GEnie- (MMD) oder Delphi (MOCKO)-Netz zu erreichen ist.

less

Textdateien unter der Lupe

AUFRUF

`less` [-dstcCeEmMuU] [-b[fp]N] [-p] [-P[mM]⟨*String*⟩] [-[lL]⟨*Logdatei*⟩] [+cmd] {⟨*Dateiname*⟩}

OPTIONEN

`s` Mehrere aufeinanderfolgende Leerzeilen werden nur als einfache Leerzeile ausgegeben.

`t` Normalerweise beginnt die Vorwärtssuche in der zweiten dargestellten Zeile. Diese Option läßt die Suche erst mit dem nächsten Screen beginnen.

`c` Gewöhnlich baut `less` den Bildschirm von unten nach oben auf. Mit dieser Option erfolgt der Textaufbau von oben nach unten.

`C` Wie `c`, jedoch wird Bildschirm vor dem Neuaufbau gelöscht.

`e` `less` wird automatisch beendet, sobald das Textende zum zweiten Mal erreicht ist.

`E` `less` wird beendet, sobald das Textende erreicht ist.

`m` Reduziert das Prompt (die farbig hinterlegte Information) auf die Prozentzahl der erreichten Textposition.

`M` Reduziert das Prompt auf einen Doppelpunkt.

`u` Backspaces ([←]) werden als druckbares Zeichen bewertet.

`U` Backspaces werden als „∧“ ausgegeben.

`xN` Setzt den Tabulatorabstand auf N Zeichen.

`[z]N` Definiert die Größe eines virtuellen Fensters in N Zeilen. Um diese Zeilenanzahl wird bei den Tastaturkommandos [Space] , [f] , [b] vor- bzw. rückwärts gescrollt.

© 1990 technicSupport

b Legt die Puffergröße in $N * 1024$ Byte fest. `less` wählt die Puffergröße abhängig davon, ob der Text aus einer Pipe (`p`) oder einer Datei (`f`) kommt. Voreingestellt sind die Werte 5 (Dateien) und 12 (Pipes). Bei der Verwendung von Pipes ist diese Option bedeutsam, da die maximal mögliche Rückwärtsbewegung durch die Größe des Puffers bestimmt wird.

p Mit dem Befehl „run less -p ⟨*Dateiname*⟩" läßt sich eine Datei ausdrucken.

P `less` kennt drei verschieden ausführliche Prompts, die sich mit der `mM`-Option anwählen lassen. Mit `P` lassen sich die drei voreingestellten Formen erweitern oder umdefinieren. Textstrings müssen in spitze Klammern eingeschlossen, die gesamte Definition muß mit einem Dollarzeichen beendet werden. Als Variable für die integrierten Prompts stehen die Strings „fo" (kurzes Prompt), „fop" (mittleres Prompt) und „fobp" (langes Prompt) zur Verfügung. Beispielsweise sorgt die Option „-P<Text heißt> fobp" beim Aufruf einer Datei namens „Spaßvogel" in der Informationszeile für eine Meldung „Text heißt Spaßvogel byte 796/2549 (29%)".

`+<Kommando>`
Wird einem Kommando ein Plus-Zeichen vorangestellt, so wird dieses Kommando sofort nach dem Aufruf ausgeführt. `+G` bewirkt einen sofortigen Sprung an das Textende, `+/Suchbegriff` läßt die Darstellung an der Textstelle beginnen, an der das Wort „Suchbegriff" zum ersten Mal auftritt. Setzt man dem Kommando zwei Pluszeichen voran, so fungiert es bei jeder in der Kommandozeile aufgeführten Datei als erster Befehl.

l Dieser Befehl fertigt gewissermaßen einen Mitschnitt der Textbetrachtung an. Der Text wird, so wie er betrachtet wird, in die angegebene Datei geschrieben. Existiert eine Datei dieses Namens bereits, so fragt `less` nach, ob die Datei wirklich überschrieben werden soll. Diese Option macht Sinn, wenn der Eingabestrom über eine `Pipe` geleitet wird.

L Wie `l`, jedoch wird auf eine Sicherheitsabfrage verzichtet.

⟨*Dateiname*⟩ Der Name der anzuzeigenden Textdatei. Es dürfen mehrere Dateien hintereinander aufgeführt werden.

ZWECK

`less` ist ein Programm, welches in erster Linie zum Anzeigen und Sichten von Texten auf dem Bildschirm dient. Weniger bekannt als das Programm selbst, das auf älteren Fishdisks die Begleittexte auf den Schirm bringt, dürften die zahlreichen Möglichkeiten sein, mit denen `less` die „Erforschung" eines Textes vereinfacht. Die neueste Version unterstützt auch Interlace-Screens und verfügt über eine Funktion zum Ausdrucken der betrachteten Texte. Textänderungen lassen sich mit `less` allerdings nicht durchführen.

BESCHREIBUNG

Von den vielen Optionen für die Kommandozeile sollte man sich nicht abschrecken lassen, sie sind nur dann erforderlich, wenn man sich über die Voreinstellungen hinwegsetzen möchte. Normalerweise genügt aus dem CLI heraus der Aufruf

`less` ⟨*Dateiname*⟩

der ein Fenster öffnet, in dem der Text angezeigt wird. Wesentlich wichtiger als die Optionen sind die Tastaturbefehle, die die Bewegung durch den Text steuern. Sie sind in der folgenden Übersicht zusammengefaßt. Viele dieser Kommandos können durch die Angabe einer Zahl N näher bestimmt werden. Diese Zahl ist einzutippen, bevor die Taste betätigt wird, die die eigentliche Funktion auslöst.

[H] Hilfsfunktion: Zeigt eine Befehlsübersicht.

[Q] `less` wird beendet.

[-] Diese Taste ist besonders wichtig, mit ihr können die unter den Optionen beschriebenen Voreinstellungen auch aus dem laufenden Programm heraus geändert werden. Wird ihr die entsprechende Option hintenangestellt, wird die zugehörige Funktion an- bzw. ausgeschaltet, abhängig von ihrem vorherigen Zustand. Mit dieser Taste lassen sich die Wirkungen der verschiedenen Optionen auch besonders einfach ausprobieren.

 © 1990 technicSupport

```
Less 1.3 by Bob Leivian: (space) for more b-ackup h-elp q-uit &print

less - opposite of more

less [-dstwcCeEnMqQuU] [-hN] [-b[fp]N] [-xN] [-[z]N]
     [-P[mM]string] [-[lL]logfile] [+cmd]  [filename]...

Less is a program similar to more (1), but which allows
backwards movement in the file as well as forward movement.
Also, less does not have to read the entire input file
before starting, so with large input files it starts up fas-
ter than text editors like vi (1). Less uses termcap, so it
can run on a variety of terminals. There is even limited
support for hardcopy terminals. (On a hardcopy terminal,
lines which should be printed at the top of the screen are
prefixed with an up-arrow.)

     Commands are based on both more and vi.  Commands may

Less.doc byte 904/15308 (5%)
```

Less: Textbetrachtung und Analyse am Bildschirm

[Space] Blättert N Zeilen vorwärts. Voreingestellt ist eine Bildschirmseite (vgl. z-Option).

[F] Wie [Space] .

[B] Blättert N Zeilen rückwärts. Voreingestellt auf eine Bildschirmseite.

[Return] Blättert N Zeilen vorwärts. Die Voreinstellung beträgt eine Zeile.

[E] Wie [Return] .

[J] Wie [Return] .

[Y] Blättert um N Zeilen rückwärts. N ist auf den Wert 1 voreingestellt.

[K] Wie [Y] .

[D] Blättert um N Zeilen vorwärts. Voreinstellung: N=10. Wird ein Wert für N angegeben, so wird dieser als neue Voreinstellung übernommen. Für [D] und [U] gilt immer eine gemeinsame Voreinstellung.

[U] Wie [D], jedoch für die Rückwärtsbewegung.

[R] Baut den Bildschirm neu auf.

[Shift]-[R]
Baut den Bildschirm neu auf, löscht dabei aber den Puffer, so daß die Daten neu eingelesen werden. Sinnvoll, wenn eine Datei während des Betrachtens von einem anderen Prozeß verändert wird.

[G] Springe zu Zeile N. Voreingestellt ist der Wert 1, also der Textanfang.

[Shift]-[G]
Gehe zu Zeile N. Voreingestellt auf das Textende.

[P] Springe zu der Stelle, die N% der Dateilänge entspricht.

[%] Wie [P].

[M] Diesem Befehl muß ein kleingeschriebener Buchstabe folgen, der als Markierung für die momentane Position dient.

['] Gefolgt von einem Kleinbuchstaben, bewirkt diese Taste einen Sprung an die entsprechende Markierung.

[/] ⟨*Suchbegriff*⟩
Sucht in Vorwärtsrichtung nach der N-ten Zeile, die den Suchbegriff enthält. N ist auf 1 voreingestellt. Die Suche beginnt in der zweiten sichtbaren Zeile, der Text wird soweit gescrollt, daß der gefundene Begriff in der obersten Bildschirmzeile erscheint.

[?] ⟨*Suchbegriff*⟩
Wie [/], jedoch in Rückwärtsrichtung. Die Suche beginnt in der ersten nicht mehr sichtbaren Zeile oberhalb des Bildschirms.

[n] Wiederholt das letzte [/] - oder [?] -Kommando.

© 1990 technicSupport

[Shift] - [E]
Das Prompt ändert sich in „Examine:“ und erlaubt die Angabe eines Dateinamens, der geladen wird. Existiert die Datei nicht, kehrt `less` zur momentanen Datei zurück. Durch Eingabe eines „#“ läßt sich jederzeit die zuvor bearbeitete Datei zurückholen.

[N] Lade die nächste in der Kommandozeile angegebene Datei. Durch Voranstellen einer Zahl können auch mehrere Dateien übersprungen werden.

[Shift] - [P]
Lade die vorhergehende Datei.

[=] Gibt einige Informationen über die Datei aus.

[Shift] - [V]
Gibt die Versionsnummer von `less` aus.

[V] Ruft einen Editor namens „Ed“ auf und veranlaßt ihn, die aktuelle Datei zu laden. Möchte man einen anderen Editor als den AmigaDOS-Ed benutzen, so kann man `Ed` einen anderen Namen geben und seinen favorisierten Editor in „Ed“ umbenennen.

[!] ⟨*DOS-Kommando*⟩
Der angegebene Befehl wird ausgeführt, Standard-Ausgaben werden im `less`-Fenster ausgegeben. „!!“ wiederholt das letzte eingegebene Kommando.

[&] Veranlaßt den Ausdruck des geladenen Textes. In den `Preferences` muß der richtige Druckertyp eingestellt sein. Vor dem Druckbeginn erscheint ein Requester, in dem man den Befehl durch Eingabe von [Q] stornieren oder mit [Return] bestätigen kann. Vor dem Drücken der [Return]-Taste lassen sich noch einige Parameter festlegen, indem ein Buchstabe eingegeben wird. Daß dieser Buchstabe zweimal auf dem Bildschirm erscheint, ist nicht von Bedeutung. Die Parameter lauten „n“ (Ausgabe mit Zeilennumerierung), „h“ (Ausgabe mit Kopfzeile, die Dateinamen und Seitenzahl enthält), „#⟨*Zahl*⟩“ (Zeilen pro Seite), „l⟨*Zahl*⟩“ (Spalten pro Zeile), „t⟨*Zahl*⟩“

(Tabulatorabstand), „∧“ (Control-Codes werden in der Form „∧⟨*Zeichen*⟩“ ausgegeben), „l“ (Schönschrift) und „d“ für EDV-Qualität. Die Parameter „P“, „w“ und „u“ hatten bei meinem P6 keine Wirkung. Da unter die Control-Codes auch die deutschen Sonderzeichen fallen, ist von der „∧“-Option abzuraten, die Umlaute werden dann nicht mehr korrekt zu Papier gebracht.

Wie die Fishdisks beweisen, lassen sich Texte auch über ein Icon anzeigen. Um dies zu erreichen muß die Textdatei mit einem Icon gleichen Namens (von der Endung „.info“ einmal abgesehen) versehen werden. Dieses Icon muß vom Typ `Project` sein, oder sein Typ muß mit einem entsprechenden Programm wie `IconType` (vgl. Band II, S. 231 f) dahingehend abgeändert werden. Anschließend ist das Icon einmal anzuklicken und die `Info`-Funktion aus dem Workbench-Menü aufzurufen. In das `DEFAULT TOOL`-Eingabefeld gibt man nun den Pfad zum `less`-Programm ein, z. B.

```
c:less
```

wenn sich das `less`-Programm im `c:`-Verzeichnis der Systemdiskette befindet.

PROBLEME

`less` hat einen Schönheitsfehler: Deutsche Sonderzeichen werden auf dem Bildschirm falsch dargestellt. Beim Sichten von Texten ist dies meist nicht tragisch, es macht jedoch einen unprofessionellen Eindruck, wenn man `less` für die Präsentation eigener Texte — beispielsweise auf Demo-Disketten — verwenden möchte.

HERKUNFT

Die hier besprochene Version 1.3 befindet sich einschließlich des Quellcodes in C auf der Fishdisk 149. Autor ist:

```
Bob Leivian
2702 W. Curry St.
Chandler, AZ 85224, USA
(602)820-6859
```

 © 1990 technicSupport

MachII

Viele Utilities in einem Programm

AUFRUF

`MachII` wird als Hintergrundprozeß aus dem CLI aufgerufen:

`Run MachII` bzw. `RunBack MachII`

Zuvor müssen die Programme `MachII.UnReg` sowie `SetMach` in ein im Suchpfad liegendes Verzeichnis, z. B. `C:`, kopiert werden. Das Hauptprogramm sollte dabei gleich umbenannt werden:

```
copy AmigaLibDisk254:MachII/MachII.UnReg C:MachII
copy AmigaLibDisk254:MachII/SetMach C:
```

Bei Verwendung einer älteren `RunBack`-Version wird beim Aufruf eine Pfadangabe erforderlich.

ZWECK

`MachII` vereinigt viele Funktionen, die ansonsten durch getrennte Utility-Programme realisiert werden, in einem einzigen Programm. Es bietet eine programmierbare Funktionstastenbelegung in drei Ebenen, die Möglichkeit, diese Tastatur-Makros selbständig zu einem bestimmten Zeitpunkt ausführen zu lassen, einen „Mausbeschleuniger", ein PopCli-Feature, einen integrierten ScreenBlanker mit getrennt programmierbaren Zeiten für Pointer und Screen, Uhr, Alarm und Gebührenzähler sowie viele weitere nützliche Features, die man bei der täglichen Arbeit am Rechner schnell nicht mehr missen will. Sein besonderer Vorteil ist aber, daß es trotz all dieser Möglichkeiten nur knapp 30 KByte lang ist und so im Vergleich zu den Programmen, die diese Möglichkeiten bisher einzeln boten (`DMouse`, `PopCli`, `FuncKey` etc.) ganz erheblich Speicherplatz spart.

BESCHREIBUNG

Komfortables Arbeiten mit dem Amiga hatte bisher seinen Preis: Wer alle Arbeitserleichterungen nutzen wollte, die ihm die PD bot,

mußte ein halbes Dutzend Programme in der Startup-Sequence starten und sich darüber hinaus noch mit den aufgrund der Menge kaum merkbaren Optionen herumärgern — ganz abgesehen von der Zeit, die beim Booten verlorenging, und dem enormen Speicherverbrauch, da jedes Programm einzeln geladen wurde. Nahezu alle der wirklich nützlichen Funktionen vereinigt `MachII`; `FuncKey`, `DMouse`, `PopCli` und einige mehr werden nun überflüssig: Wer einmal mit `MachII` arbeitete, möchte es nicht mehr missen!

Nachdem `MachII` gestartet wurde (was nur vom CLI aus möglich ist und am besten schon in der Startup-Sequence geschehen sollte), erscheint ein kleines Fenster, das Informationen über das Programm zeigt. Nur beim ersten Start von `MachII` macht vorher ein Requester darauf aufmerksam, daß die Konfigurationsdatei nicht gefunden werden konnte. Er kann jedoch durch Anklicken von [Cancel] entfernt werden.

Aktiviert man nun das Fenster von `MachII` und drückt anschließend die rechte Maustaste, so erscheint der Requester zum Einstellen der Optionen. Diese werden auf ähnliche Weise wie in den `Preferences` der Workbench festgelegt und gespeichert. Alternativ gibt es noch eine spezielle Tastenkombination, um diesen Requester aufzurufen: [Ctrl] -[Amiga] -[Esc] tut das gleiche.

In den obersten drei Reihen dieses Requesters werden die Funktionstasten definiert. Das linke Gadget der ersten Reihe dient zum Umschalten der drei Ebenen, in denen die Tasten definiert werden können: [NONE] steht für die Funktionstaste alleine, [Amiga] für die Funktionstaste zusammen mit einer beliebigen Amiga-Taste und [CTL-Amiga] für die F-Taste zusammen mit [Ctrl] und einer der beiden Amiga-Tasten. Das [c/ce] -Gadget ist mit Vorsicht zu genießen: Es löscht das momentane Makro, bei einem Doppelklick jedoch (ohne Sicherheitsabfrage) alle definierten Makros. Im `Size`-Gadget kann schließlich die maximale Länge der Definitionen eingestellt werden. Voreingestellt ist hier 256. In der Reihe darunter kann die zu belegende F-Taste angewählt werden, und in der dritten Reihe findet sich das String-Gadget, das den Text aufnimmt.

 © 1990 technicSupport

Die bereits kurz erwähnte Alternative zur Funktionstastenbelegung in dieser Weise scheint zunächst umständlicher, ist nach kurzer Gewöhnungszeit jedoch um einiges bequemer: `MachII` kennt sogenannte „Lern-Sequenzen". Drückt man die Spezial-Kombination [Alt]-[Esc], so beginnt für `MachII` eine solche Sequenz. Es fängt nun alle Tastatureingaben bis zum erneuten [Alt]-[Esc] ab und erwartet zunächst die Eingabe der entsprechenden Taste und danach ihre Belegung. Die Maus sollte dabei tunlichst in Ruhe gelassen werden, denn alle Eingabeereignisse werden registriert.

Vollziehen wir dies an einem einfachen Beispiel: Die Tastenkombination [Ctrl] - [Amiga] -[F10] soll mit dem Befehl `FORMAT DRIVE DF1: NAME Empty NOICONS` belegt werden.

Wir drücken dazu gleichzeitig [Alt]-[Esc], auf die Aufforderung „*Select Hotkey...*" dann (wieder gleichzeitig) die zu belegende Tastenkombination [Ctrl] - [Amiga] -[F10] und geben dann auf die Aufforderung „*Enter Macro...*" hin unseren Text ein, wobei wir uns das abschließende [Return] nur aus Sicherheitsgründen schenken, `MachII` würde es ohne Probleme verarbeiten. Ein erneutes [Alt]-[Esc] sagt `MachII`, daß die Definition abgeschlossen ist. Das Löschen ist ebenso einfach: [Alt]-[Esc], [Ctrl] -[Amiga] -[F10] und nochmals [Alt]-[Esc], und `MachII` hat die Definition vergessen und behandelt diese Tastenkombination wieder normal: Nicht belegte Kombinationen liefern unter `MachII` keinen Leerstring, sondern die ganz normalen Steuerzeichen-Sequenzen, die von anderen Programmen teilweise erwartet werden.

In der vierten Zeile kann der Befehlsstring zum Aufruf eines neuen CLI's mit der Spezial-Kombination [Amiga]-[Esc] definiert werden, per Default lautet er `NewCLI`, was aber auch in `NewShell` geändert werden kann, um beispielsweise statt des CLI eine Shell zu starten. Der Unterschied zwischen dieser ganz speziellen Tastenbelegung und den normalen Makros ist der, daß der hier eingegebene String nicht einem CLI-Fenster, sondern der DOS-Funktion Execute() übergeben wird und so auch ohne ein geöffnetes CLI-Fenster ausgeführt werden kann. Denkbare Alternativen wären z. B. der Aufruf von `VIEW` (auf der Fish 163 zu finden), um beim Arbeiten mit einem PD-Programm bei Bedarf auf das Doc-File zugreifen zu können,

oder der Aufruf von `Tetrix` (Fishdisk 173), um bei aufkommendem Streß ein wenig Entspannung bei einem Spielchen zu suchen...

Zu beachten ist nur folgendes: Der für diese Aufrufe erforderliche `Run`-Befehl wird unter `C:Run` gesucht und muß auch dort stehen, Pfadumleitungen mit Path sind nutzlos. Der Stack hat dieselbe Größe wie der des CLI's, aus dem `MachII` gestartet wurde und muß ggf. vergrößert werden.

Rechts oben finden sich dann die Gadgets, die das Fensterhandling steuern. Mit dem [SunMouse] -Gadget läßt sich das `SunMouse`-Feature ein- oder ausschalten. Ist es aktiviert, ist das Fenster, über dem der Mauszeiger steht, immer auch das aktive Fenster. Daneben läßt sich die Zahl der Mausklicks einstellen, die erforderlich sind, um ein Fenster in den Vordergrund zu bringen. `1` steht für Einzelklick, `2` für Doppelklick und bei `0` ist die Funktion desaktiviert, so daß nur noch das Depth-Gadget des Fensters zur Verfügung steht. Mit dem [Cycle] -Gadget ist ein weiteres interessantes Feature zuschaltbar: Bei aktiviertem Cycle läßt sich ein Fenster durch Drücken des rechten Mausknopfes bei gleichzeitig gedrückt gehaltenem linkem Mausknopf in den Hintergrund verfrachten, was sich komplizierter anhört, als es ist. Es handelt sich um dieselbe Tastenfolge wie bei der Mehrfachbetätigung von Intuition-Menüs. Unter `Accel` (für *Acceleration*) läßt sich der Beschleunigungsfaktor für den Mauszeiger festlegen. 0 steht für normale Bewegung, 9 für maximale Beschleunigung. Brauchbare Werte sind 1 oder 2. `MouseOff` legt die Anzahl der Sekunden fest, nach denen der Mauszeiger verschwindet, wenn die Maus nicht benutzt wird.

Das [Clock] -Gadget schaltet das immer sichtbare `MachII`-Fenster mit der Uhrzeit- und Speicherplatzanzeige wahlweise aus. [Hrs] wechselt zwischen 12- und 24-Stunden-Anzeige. Das `Meter`-Gadget ersetzt die Uhrzeitanzeige durch einen Gebührenzähler, der DFÜ-Fans sicher nützlich sein wird. Die Eingabe des Gebührensatzes erfolgt beim `Rate`-Gadget in amerikanischen Cents pro Stunde, läuft aber natürlich auch in Pfennigen pro Stunde. Bei einer normalen Telefonverbindung im Ortstarif nach 18 Uhr wäre hier also 105 einzutragen (für 5 Einheiten), bei einer Verbindung der Zone 3 vor 18 Uhr 6300 (für 300 Einheiten). Der Zählerstand wird jede Minute aktualisiert.

 © 1990 technicSupport

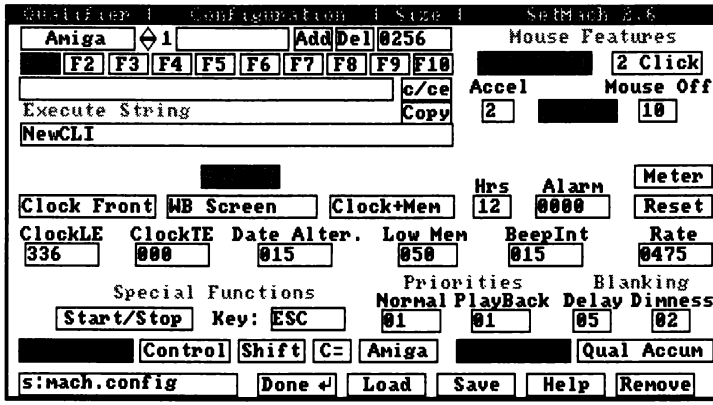
Mit dem [Reset] -Gadget wird der Zähler gleichzeitig auf Null gesetzt und neu gestartet.

Mit den drei Gadgets in der siebten Zeile läßt sich Aussehen und Verhaltensweise des Uhr-Fensters bestimmen. Das linke schaltet zwischen [Clock Front] und [Clock Depth] um, ersteres hält die Uhr immer vor anderen Fenstern, letzteres fügt der Uhr zwei Depth-Gadgets hinzu, mit denen das Fenster in gewohnter Weise gehandhabt werden kann. Das mittlere Gadget hat drei Zustände: [WB Screen] hält die Uhr immer auf dem Workbench-Screen, [Screen Depth] erlaubt, den Screen der Uhr mittels eines unsichtbaren zusätzlichen Gadgets am linken Rand der Uhr zu wechseln, und [Clock to Top] bewegt die Uhr automatisch auf den vordersten Screen. Das rechte Gadget erlaubt schließlich die Auswahl zwischen reiner Uhrzeitanzeige, reiner Speicheranzeige oder beidem.

In der Zeile darunter können bei `Clock Left Edge` und `Clock Top Edge` die Koordinaten der Uhr eingestellt werden (geht einfacher, indem die Uhr an die richtige Stelle geschoben wird und dann erst der Requester aufgerufen wird), bei `Date Alter.` das Intervall der Datumseinblendung in Sekunden und bei `Rate` die Kosten für eine Online-Verbindung in Cents/Stunde (bzw. Pfennigen/Stunde, s.o.) eingegeben werden. Im [Low-Mem] -Gadget läßt sich der freie Speicher einstellen, bei dessen Unterschreitung die Anzeige die Farbe wechselt. Der Default 50 ist ein brauchbarer Wert.

Bei `Beep Int` kann die Zeit in Minuten eingestellt werden, die zwischen den Beeps der Uhr liegt. Bei Eingabe von 60 beept die Uhr stündlich, bei 15 (Default) viertelstündlich. 0 desaktiviert die Funktion. Das `Alarm`-Gadget nimmt die Alarm-Uhrzeit zwischen 0001 und 2359 auf. Werte darüber schalten den Alarm aus. (Voreinstellung: 2500 für keinen Alarm).

Die beiden Gadgets unter `Blanking` enthalten die Optionen des Screen-Blankers: Links die Zeit in Minuten bis zum Ausblenden des Screens, in der Mitte die Stärke der Verdunkelung (0 = schwarzer Bildschirm, 15 = keine Abblendung) und rechts die Zeit in Sekunden bis zum Ausblenden des Mauszeigers. Eine 0 desaktiviert das Feature.



MachII vereinigt viele Utilities in einem Programm

In den nächsten beiden Gadget-Leisten lassen sich die Spezial-Tastenkombinationen von `MachII` umdefinieren. Das erste der Gadgets bestimmt, welche der vier Kombinationen gerade umdefiniert wird. [Start/Stop] bezeichnet die Kombination zum Einleiten der Lern-Sequenzen, [Execute] die Kombination für die PopCLI-Fähigkeit, [SetMach] die Kombination zum Aufruf des Einstellrequesters und [Matcher] die Kombination zum Aufruf des Matchers, der wenn aktiviert, eine neue Möglichkeit des Makroaufrufs darstellt: Man gibt solange Zeichen ein, bis das zugehörige Makro eindeutig definiert ist, bei dem obigen FORMAT-Beispiel also nur ein `F`, wenn es das einzige mit `F` beginnende Makro ist. Die voreingestellten Kombinationen erscheinen mir jedoch recht sinnvoll. Mit den restlichen Gadgets läßt sich die Tastenkombination definieren; es stehen insgesamt 38 Tasten zur Verfügung, die mit den [Alt] -, [Ctrl] -, [Shift] - und [Amiga] -Tasten noch vervielfacht werden können, so daß sich wohl immer eine Kombination finden läßt, die das jeweilige Anwendungsprogramm noch nicht belegt hat. So kann allen Kompatibilitätsproblemen elegant ausgewichen werden.

 © 1990 technicSupport

Das [Amiga-M] -Gadget erlaubt es, wenn aktiviert, mit der Tastenkombination [Amiga] - [M] nicht nur zwischen zwei, sondern der Reihe nach zwischen allen Screens umzuschalten.

Das [Qual Accum] -Gadget schaltet den Qualifier Accumulator-Modus ein. Das bedeutet, daß alle Sondertasten ([Shift] , [Ctrl] , [Alt] und die beiden Amiga-Tasten) solange gespeichert werden, bis die nächste Nicht-Sondertaste gedrückt wird. Um also beispielsweise die Tastenfolge [Amiga] - [/] einzugeben, muß [Amiga] -[Shift] -[7] nicht mehr gleichzeitig gedrückt werden, sondern die Tasten können auch einzeln nacheinander gedrückt werden. Ein nützliches Feature für Leute mit kurzen Fingern.

Im Text-Gadget ganz unten links kann der Name der Konfigurationsdatei editiert werden, um mehrere solcher Dateien anlegen zu können. Beim ersten Start lädt `MachII` aber *immer* aus `s:Mach.-config`. Unter anderen Namen abgelegte Config-Dateien müssen danach von Hand geladen werden.

In der letzten Zeile schließlich kann die Konfigurationsdatei geladen und gespeichert sowie der Requester mit [Done] oder [Remove] beendet werden. [Done] schließt nur den Requester, [Remove] entfernt `MachII` ganz aus dem Speicher. Insbesondere bei der Makro-Definition bietet `MachII` noch viele zusätzliche Features durch Steuerzeichen, auf die hier aus Platzgründen nicht eingegangen werden kann. Ein Hilfsfenster mit Esc-Sequenzen läßt sich Anklicken von [Help] oder Drücken der [Help] -Taste aufrufen. Interessierte seien auf die englische Anleitung verwiesen.

Eine Konfigurationsdatei kann jedoch nicht nur eine, sondern mehrere Einstellungen speichern, wozu die bisher vernachlässigten Gadgets der obersten Leiste dienen. Verträgt sich zum Beispiel die gewünschte Belegung der Funktionstasten nicht mit einem bestimmten Programm, dann kann man durch Anlicken von [Add] eine zweite identische Einstellung erzeugen (durch mehrere Einstellungen bewegt man sich durch Anklicken der [▽] - und [△] -Symbole). Rechts neben diesen Symbolen befindet sich ein Textgadget, in das man, nachdem man für die zweite Einstellung die Funktionstasten gelöscht hat, den Anfang eines Fenstertitels eingeben, zum Beispiel

E für Ed. Immer wenn man in einem Fenster arbeitet, dessen Titel mit diesen Buchstaben beginnt, aktiviert MachII automatisch die zweite Einstellung.

PROBLEME

Probleme tauchen in Zusammenhang mit dem PopCLI-Feature auf, wenn der Befehl c:Run fehlt. Mit den Möglichkeiten zum Arrangieren von Fenstern sollte vorsichtig umgegangen werden, da Intuition zu schnell aufeinanderfolgende Größen oder Positionsänderungen nicht immer korrekt verarbeitet, was zu einem Guru führen kann. Führt man die entsprechenden Aktionen gemächlich durch, besteht keine Gefahr. Hektisches Herumspielen ist aber gefährlich!

HERKUNFT

MachII findet sich in der hier besprochenen Version 2.6 auf der Fishdisk 254. Das Programm ist Shareware. Für $15 (Europäer sollten etwas fürs Porto drauflegen) wird man registrierter Anwender und erhält eine noch kürzere Version. Die Adresse des Autors:

```
Brian Moats
Polyglot Software
10431 Ardyce Court
Boise, Idaho 83704
```

Über PeopleLink ist er als BrianM zu erreichen, über CompuServe unter 76167,2334.

VERWEISE

DMouse (Fishdisk 258), FuncKey (Fishdisk 106), PopCLI (Fishdisk 84), Mackie (Fishdisk 267)

 © 1990 technicSupport

Mackie

Bildschirmschoner mit Extras

AUFRUF

`Mackie` wird aus dem CLI aufgerufen:

[`run`] `mackie` [-q] [-l] [-b] [-f ⟨*Dateiname*⟩] [⟨*Kommando*⟩] [⟨*Zeit*⟩]

OPTIONEN

`-q`	Entfernt einen bereits installierten `Mackie`-Task aus dem Speicher.
`-l`	Anstatt des schwarzen Bildschirmes sollen Linienmuster erscheinen (Grundeinstellung)
`-b`	Anstatt der Linienmuster wird ein schwarzer Bildschirm gezeichnet.
`-f`	Nach dieser Option folgt der Pfad/Name der Konfigurationsdatei. Voreingestellt ist der name `s:.mackierc`
`<Kommando>`	Hier steht der auf die Tastenkombination [linke Amiga] -[Esc] hin auszuführende Befehl.
`<Zeit>`	Die Zeit in Sekunden, die ohne Benutzeraktivität verstreichen muß, bevor der Bildschirm auf Dunkel oder Liniendarstellung schaltet.

ZWECK

Viele Hilfsprogramme haben sich schon des Problems angenommen: Da steht man mal kurz auf, vergißt, daß Rechner und Monitor eingeschaltet sind, und geht aus dem Haus. Geschieht dies öfters, kann es geschehen, daß die Phosphor-Schicht des Monitors durch ein sich einbrennendes Bild zerstört wird. Um diesem Umstand vorzubeugen, schalten bestimmte Programme den Bildschirm nach einer frei wählbaren Zeit, in der keinerlei Maus- oder Tastaturaktivitäten stattfinden, schwarz. So wird der Monitor geschont.

Mackie hingegen schaltet den Bildschirm in einem solchen Fall nicht komplett schwarz, sondern malt verschiedene, sich verändernde Linienmuster, deren Grundstruktur alle fünf Minuten wechselt, in hoher Geschwindigkeit auf den Schirm. Eine sehr ästhetische Methode, das Einbrennen eines Standbildes zu verhindern! Doch Mackie kann noch mehr. Zusätzlich kann die [linke Amiga] -Taste in Verbindung mit [Esc] einen beim Aufruf des Programms angegebenen Befehl jederzeit ausführen. Normalerweise ist dies der Aufruf eines neuen CLI- oder Shell-Fensters.

Doch der Funktionen nicht genug. Als erstes Programm definiert Mackie die [Help] -Taste so um, daß bestimmte Programme auf Tastendruck gestartet werden — egal ob diese sich schon im Speicher befinden (dann ploppt der Bildschirm nur nach vorne) oder nicht: Dann wird das Programm neu eingeladen. Das geht, indem man die [Help] -Taste einmal drückt und wieder losläßt, um danach den Anfangsbuchstaben des Fenster- bzw. Bildschirmtitels des entsprechenden Programmes einzugeben. Diese Möglichkeit steht allerdings nur Anwendern zur Verfügung, die statt der AmigaShell die kommerzielle WShell benutzen.

BESCHREIBUNG

Die einfachste Möglichkeit, Mackie zu starten, ist aus dem Aufruf ersichtlich. Hier werden die entsprechenden Optionen (bis auf -f) angegeben. Beispiel:

```
run mackie "newshell con:0/10/580/100/MackieShell" 120
```

Damit wird bei jedem Druck auf [linke Amiga] -[Esc] eine neue Shell mit den angegebenen Koordinaten aufgerufen. Nach einer Eingabepause von mehr als 2 Minuten wird auf Liniendarstellung (-l-Option ist voreingestellt!) umgeschaltet. Um Mackie wieder zu entfernen, genügt der Befehl:

```
mackie -q
```

Doch Mackie kann wesentlich mehr. Die zusätzlichen Features lassen sich jedoch nur über eine Konfigurationsdatei nutzen. Als erläutern-

 © 1990 technicSupport

des Beispiel ist hier eine Konfigurationsdatei aufgeführt:

```
;Für User mit WShell und ConMan
command "NewWSH con:0/100/640/80/MeinCLI/c"
;Nach 2 Minuten dunkel schalten
timeout 120
;Sondertasten definieren
c CLI "" "newwsh"
x INTUITION "xoper*" "xoper"
p INTUITION "preview*" "dh2:tex/c/preview"
SHIFT t CLI "tex*" "newwsh from s:texsetup"
w INTUITION ""
```

Die ersten 4 Zeilen sind nicht schwer zu verstehen. **Command** ist identisch mit der Angabe von ⟨*Kommando*⟩ beim Aufruf ohne Konfigurationsdatei, und **timeout** ist identisch mit ⟨*Zeit*⟩. Alle Zeilen, die mit einem Semikolon beginnen, werden von **Mackie** als Kommentarzeilen angesehen und dementsprechend ignoriert.

Aber auch die Definition der Sondertasten ist nicht schwierig. Diese Tasten — es können alle Buchstaben, Zahlen und Funktionstasten verwendet werden — führen den angegebenen Befehl aus, wenn zuerst die [Help] -Taste und dann die entsprechende Sondertaste gedrückt wird. Der Befehl wird jedoch nicht ausgeführt, wenn ein Fenster oder Bildschirm mit dem angegebenen Namen (* dient als Joker für beliebige Zeichen) bereits existiert. In diesem Falle wird das Fenster oder der Bildschirm lediglich nach vorne geholt und aktiviert. Um auch die [Help] -Taste noch eingeben zu können, muß diese einfach zweimal gedrückt werden.

Die Vorgehensweise, nach der ein Programm im Speicher gesucht, gefunden oder neu eingeladen wird, ist am besten an einem Beispiel zu demonstrieren: Drückt man nach der Konfiguration von **Mackie** mit obiger Datei [Help] und danach [C] , sucht **Mackie** ein CLI-Fenster, das im Moment keinen Befehl ausführt (momentaner Name = ""). Falls dieses gefunden wird, ploppt das Fenster automatisch nach vorn, falls nicht, wird der nachfolgende Befehl **newwsh** ausgeführt, der wiederum ein neues CLI-Fenster zur Verfügung stellt.

Genauso verhält es sich mit dem Systemmonitor **Xoper**. Ist dieser geladen und hat sein Fenster geöffnet (ein Task, der mit den Buchstaben **xoper** beginnt, deshalb das nachfolgende Joker-Sternchen), wird nur das Fenster nach vorne geholt. Ansonsten wird auch noch der letzte Parameter ausgeführt, und der heißt hier **xoper** und lädt somit den Monitor in den Speicher.

Die weiteren Definitionen dürften für sich sprechen. Zum einen wird der AmigaTEX-Textvorbetrachter **Preview** gesucht und unter Umständen eingeladen, zum anderen eine Stapeldatei in einem eigenen CLI-Fenster gestartet. Beachten sollte man das Schlüsselwort **Shift**, das besagt, diesen Befehl nur auszuführen, wenn nach [Help] auch [Shift] und die Sondertaste gedrückt wird. Funktionstasten erhalten übrigens die Bezeichnung **Fx** wobei **x** von 1 bis 10 geht. Die Leertaste heißt **space**. Alle in der Konfigurationsdatei nicht definierten Tasten erhalten die Definition:

```
x "x*"
```

Dies bedeutet, daß sie nach Fenster oder Bildschirmen suchen, die mit dem angegebenen Buchstaben beginnen, und diese nach vorne holen — egal, ob es sich dabei um ein CLI- oder Intuition-Fenster handelt. Beachten sollte man jedoch, daß nur Bildschirme nach ihrer tatsächlich erscheinenden Titelzeile gesucht werden. Bei Fenstern wird nach dem Namen des zugehörigen Tasks gesucht, da Titelzeilen von Fenstern oft zu Statusausgaben oder für Fehlermeldungen verwendet werden. Die nach obigen Muster definierten Zahlen holen in Kombination mit der [Help] -Taste übrigens das CLI-Fenster der entsprechenden Prozeßnummer nach vorne. Wurde **space** nicht definiert, schaltet [Help] - [Leertaste] alle Fenster und Bildschirme der Reihe nach durch.

PROBLEME

Um **Mackie** auch von der **Startup-Sequence** aus starten zu können, sollte statt **run** der Befehl **runback**, beispielsweise von Fishdisk 152, benutzt werden.

 © 1990 technicSupport

HERKUNFT

Die hier besprochene Programmversion 1.3 befindet sich auf Fishdisk 267. `Mackie` ist Public-Domain, und der Autor fordert keinerlei finanziellen Obolus für die Benutzung des Programmes. Man erreicht TOMAS ROKICKI unter

```
Tomas Rokicki
Box 2081
Stanford, CA 94309
```

VERWEISE

`PopCli` (Fishdisk 84), `MachII` (Fishdisk 254)

MandelVroom

Apfelmännchen-Generator

AUFRUF

`MandelVroom` läßt sich von der Workbench aus starten, zuvor ist jedoch in einem CLI-Fenster eine `Assign`-Zuweisung durchzuführen. Das Verzeichnis, in dem sich `MandelVroom` befindet, muß dabei als logisches Gerät mit dem Namen `Mandelbrot:` deklariert werden. Für die Fishdisk lautet das Kommando:

```
assign Mandelbrot: AmigaLibDisk215:MandelVroom
```

Startet man das Programm aus dem CLI heraus, muß zuvor der Stack auf einen Wert von mindestens 30.000 gesetzt werden:

```
stack 30000
```

`MandelVroom` setzt voraus, daß sich im `Libs:`-Verzeichnis die Libraries `Mathtrans.library` und `Mathieeedoubbas.library` befinden.

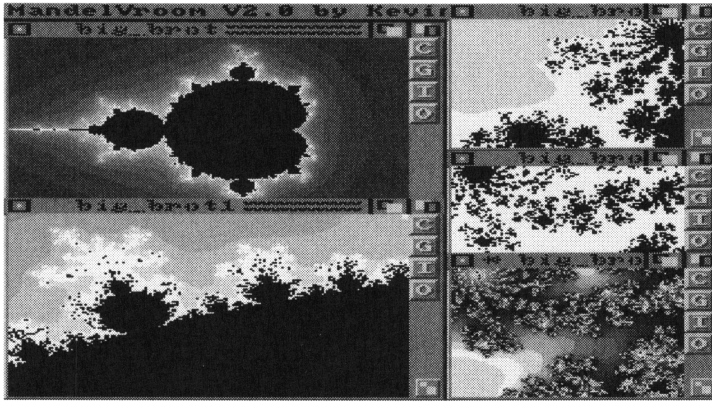
ZWECK

`MandelVroom` berechnet komplexe Zahlen aus der Mandelbrot- bzw. der Julia-Menge und stellt sie auf dem Bildschirm dar. Nüchterne Mathematik als Ausgangspunkt für anspruchsvolle Computergrafiken, die vielen Anwendern unter der Bezeichnung *Apfelmännchen* bekannt sein dürften!

BESCHREIBUNG

Um den Appetit auf `MandelVroom` nicht gleich durch zuviel graue Theorie zu verderben, wagen wir zur Abwechslung einmal den unvorbereiteten Sprung in das Programm. Hat man es, wie oben beschrieben, gestartet, öffnet sich ein neuer Screen, der ein kleines Fenster zeigt, in dem ohne weiteres Zutun Zeile für Zeile ein Bild aufgebaut wird. Diesen Vorgang sollte man zunächst in aller Ruhe

 © 1990 technicSupport



Erste Schritte mit MandelVroom

abwarten. Anschließend kann man das Fenster an der rechten unteren Ecke greifen und es vergößern. Das Bild selbst behält seine ursprüngliche Größe bei; für die neue Fenstergröße muß es neu berechnet werden. Dieser Vorgang wird durch Anklicken des [G] -Gadgets (G = *generate* — erzeugen) an der rechten Fensterseite in Gang gesetzt. Bereits jetzt wird deutlich, daß die Berechnung eines solchen Bildes sehr zeitaufwendig ist. Zum Trost: Ähnliche Programme gab es schon für den Commodore 64; damals mußte man auf ein solches Bild bisweilen noch eine ganze Nacht warten.

Neue Bilder gewinnt man, indem man einen Ausschnitt aus einem bereits vorhandenen Bild herausschneidet und diesen neu berechnen läßt. Dazu benötigt man ein weiteres Fenster, welches man nach Anwahl des Menüpunktes `Project New/Mand` erhält. Meist verdeckt dieses Fenster das alte, so daß es nötig wird, das neue zu verschieben. Nach Anklicken des [I] -Gadgets im neuen Fenster zeigt der Mauszeiger den Schriftzug „To“ und ermöglicht die Wahl eines Bildausschnittes. Man klickt mit der linken Maustaste auf einen Punkt des

alten Bildes und zieht den Mauszeiger bei immer noch gedrückter Maustaste nach rechts unten, bis man den gewünschten Ausschnitt bestimmt hat. Dieser Ausschnitt wird dabei vom neuen Fenster angezeigt. Nach dem Loslassen der Maustaste wird der Ausschnitt von einem Rahmen eingeschlossen, der große Ähnlichkeit mit einem Amiga-Fenster aufweist. Und in der Tat kann man den Rahmen in der „Titelleiste“ greifen, um nachträglich seine Position zu korrigieren. Über die rechte untere Ecke läßt sich der Ausschnitt nach rechts unten vergrößern. Das Gadget an der linken unteren Ecke sorgt für eine symmetrische Vergrößerung um den Mittelpunkt des Ausschnitts herum. Durch Anklicken der linken oberen Ecke kann die Markierung entfernt werden. Übrigens: Je mehr Schwarzanteile der Ausschnitt aufweist, umso länger wird die Berechnung des neuen Bildes dauern.

Diese Berechnung wird wieder durch Anklicken des [G] -Feldes gestartet. Dieses Verfahren läßt sich fortsetzen, indem man ein weiteres Fenster öffnet, es mit einem Ausschnitt aus Bild 2 füttert usw. Die erzeugten Bilder werden aber immer weniger Farben enthalten, was mit dem Menüpunkt `Display/Autocontour` ausgemerzt werden kann. Hier hat man zwei Möglichkeiten (`I` und `II`) zur Auswahl. Die rechten unteren beiden Bilder der vorangehenden Abbildung etwa sind auf identische Weise erzeugt worden, das untere wurde jedoch nachträglich mit `Autocontour` eingefärbt.

Das Programm

Bevor wir uns nun endgültig in die Theorie stürzen, sollten wir einen Blick auf die Eigenschaften des Programms werfen, angesichts derer sich viele kommerzielle Programme im hintersten Winkel des Speichers verstecken müßten.

- Volle Multitasking-Unterstützung
- Komplette Benutzerführung über Menüs und Gadgets, daneben Tastatur-Shortcuts
- Unterstützung von LoRes-, HiRes-, MedRes-, Interlace- und Overscan-Screens. Zwischen diesen Formaten kann jederzeit gewechselt werden.

 © 1990 technicSupport

- Unterstützung des Extra-Halfbrite-Modus
- Unterstützung der PAL-Auflösung !
- Ausnutzung des 68020/68881
- Fünf Arithmetik-Varianten: 32-Bit (für 68000 bzw. 68020), Amiga-FFP, IEEE Floating Point und IEEE Floating Point für 68881 in Assembler.
- Laden und Speichern von Projekten
- Abspeichern als IFF-Bild

Grundlagen

Auch wenn kein Hintergrundwissen erforderlich ist, um sich an den Grafiken dieses Programms zu erfreuen, ist es doch nützlich, wenigstens die grundsätzlichen Prinzipien zu kennen, nach denen `MandelVroom` arbeitet. Das Verständnis dieser Grundlagen erfordert allerdings nicht unerhebliche mathematische Kenntnisse.

Jedes Bild, das von `MandelVroom` erzeugt wird, besteht aus einer unterschiedlichen Anzahl von Bildpunkten — abhängig davon, welche Bildschirmauflösung eingestellt worden ist. Jeder dieser Bildpunkte repräsentiert einen Zahlenwert, der durch die horizontale und vertikale Position des Bildpunktes bestimmt wird. Man kann sich ein solches Bild auch von einem Koordinatensystem unterlegt vorstellen; dieses Koordinatensystem wird von `MandelVroom` aber nicht grafisch dargestellt.

Die Hauptaufgabe von `MandelVroom` besteht darin, für jeden Bildpunkt das Verhalten einer Zahlenfolge zu untersuchen, das wiederum von dem Zahlenwert des Bildpunktes abhängt. Das Verhalten dieser Zahlenfolge bestimmt schließlich die Farbwerte, die den jeweiligen Pixeln zugeordnet werden.

Die verschiedenen Verhaltensweisen, die solche Zahlenfolgen an den Tag legen kann, läßt sich an einem noch recht einfachen Beispiel diskutieren:

$$x_{n+1} := x_n^2 + c$$

Aus dieser Definition ergeben sich verschiedene Zahlenfolgen, abhängig von der Konstanten c und abhängig davon, welcher Anfangswert x_0 gewählt wird. Untersuchen wollen wir diese Folgen für den denkbar einfachsten Fall, bei dem die Konstante c den Wert Null erhält. Dann gilt beispielsweise für den Anfangswert $x_0 := 2$:

$$x_1 = x_0^2 = 2^2 = 4$$

$$x_2 = x_1^2 = 4^2 = 16$$

$$x_3 = x_2^2 = 16^2 = 256$$

Für diesen Anfangswert ergibt sich also die Folge $2, 4, 16, 256, \ldots$. Allgemeiner läßt sich schreiben:

$$x_1 = x_0^2$$

$$x_2 = x_1^2 = \left(x_0^2\right)^2 = x_0^4$$

$$x_3 = x_2^2 = \left(x_0^4\right)^2 = x_0^8$$

Man erkennt, daß der Wert für x_n ausschließlich von x_0 abhängt, egal, wie groß man den Wert n wählt. Für diesen einfachen Fall ist daher eine explizite Darstellung möglich:

$$x_n = \left(x_0^2\right)^n := x_0^n \cdot x_0^n$$

Von Interesse ist nun das Verhalten dieser Zahlenfolgen bei größer werdendem n. Abhängig von dem Ausgangswert (x_0) lassen sich drei Fälle unterscheiden:

$x_0 < 1$ Bei einem Wert, der kleiner ist als Eins, wird x_n mit wachsendem n immer kleiner und nähert sich immer mehr dem Wert Null an. Beispiel: Bei einem Wert von $\frac{1}{2}$ gilt $x_1 = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$, $x_2 = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{16}$ usw. Dasselbe gilt für negative Werte.

$x_0 > 1$ Dies ist der umgekehrte Fall: Mit wachsendem n wird x_n immer größer. Dasselbe gilt für negative Werte.

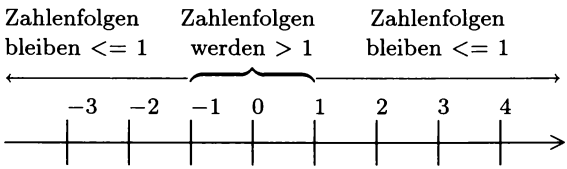
 © 1990 technicSupport

$x_0 = 1$ In diesem letzten Fall behält die Zahlenfolge immer denselben Wert von Eins. Dies ist leicht zu verstehen. Egal wie oft man die Zahl Eins mit sich selbst multipliziert: Das Ergebnis bleibt immer die Eins selbst. Das gilt auch für den Wert -1, da als Ergebnis nur geradzahlige Potenzen von x_0 auftreten.

Hinsichtlich ihrer Auswirkungen auf das Verhalten der Zahlenfolgen lassen sich die Anfangswerte x_0 unter Verwendung eines anderen Kriteriums aber auch in lediglich zwei Mengen unterscheiden.

- Die Menge der Anfangswerte x_0, für die die Zahlenfolge einen bestimmten Grenzwert (in diesem Falle 1) nicht überschreitet.
- Die Menge der Anfangswerte x_0, für welche die Zahlenfolge diesen Grenzwert überschreitet.

Für unser Beispiel läßt sich das auf dem berühmten Zahlenstrahl folgendermaßen darstellen:



Auf unserem Bildschirm sehen wir allerdings keine eindimensionale, sondern eine zweidimensionale Darstellung. Wir können uns einmal vorstellen, daß in der Mitte eines `MandelVroom`-Fensters der Nullpunkt (0|0) eines Koordinatensystems sitzt und jedem Punkt der Zahlenwert x_0 zugeordnet wird, der seinem Abstand von diesem Nullpunkt entspricht. In unserem obigen Beispiel würde die Punktmenge, für welche die Folge beschränkt bleibt, dann aus all jenen Punkten bestehen, die auf einem Kreis mit dem Radius 1 oder innerhalb dieses Kreises liegen.

Nun hat ein simpler Kreis aber nicht gerade viel mit den bizarren Grafiken gemeinsam, wie sie uns `MandelVroom` präsentiert. Der Grund: Wir haben ein sehr einfaches Beispiel gewählt, weil man das

Verhalten der oben gewählten Zahlenfolge noch recht einfach „zu Fuß“ untersuchen konnte. Die Zahlenfolgen, mit denen `MandelVroom` arbeitet, sind in ihrem Verhalten wesentlich komplizierter und erlauben keine explizite Darstellung mehr. Aus diesem Grunde setzte die intensive Erforschung dieser Gebilde auch erst im letzten Jahrzehnt ein, nachdem die ersten Computer zur Verfügung standen, denen die Rechenarbeit überlassen werden konnte.

Die Menge der komplexen Zahlen

Ohne es mit der Akribie, die ein Mathematiker verlangen würde, hinter die Formeln zu setzen, sind wir im oberen Beispiel wie selbstverständlich davon ausgegangen, daß wir für die Zahl n nur natürliche Zahlen (0, 1, 2 usw.) und für x_0 nur reelle Zahlen einsetzen. Es war BENOÎT B. MANDELBROT, der 1980 auf die Idee kam, es einmal mit komplexen Zahlen zu versuchen, und auch als erster Computer einsetzte, um die Zahlenmenge darzustellen, die kurze Zeit später nach ihm benannt werden sollte.

Komplexe Zahlen sind leider keine ganz triviale Angelegenheit. Im Rahmen dieses Buches können wir daher auch nicht ausführlich darauf eingehen, wie man mit diesen Zahlen rechnet. Wir können lediglich erklären, was eine komplexe Zahl überhaupt ist, und wie man sie darstellen kann.

In der Mathematik werden verschiedene Zahlenmengen unterschieden. Das Verständnis für die einfacheren dieser Mengen erwerben wir mehr oder weniger intuitiv im Verlaufe unserer Kindheit. Es fängt mit den natürlichen Zahlen (1, 2, usw.) an, mit denen wir unsere Finger zählen oder die Kerzen der Geburtstagstorte; wir rechnen zwei Äpfel und noch einmal zwei Äpfel zusammen und kommen auf Apfelkuchen, wir ziehen drei Äpfel ab und bekommen Bauchschmerzen. Mit dem Aufteilen der Torte lernen wir die Bruchzahlen kennen: Ein Achtel für Mama, ein Achtel für Papa, der Rest für mich. Irgendwann brauchen wir einen Taschengeldvorschuß und werden mit den negativen Zahlen konfrontiert. Probleme bekommen die Lehrer spätestens dann, wenn sie uns erklären müssen, daß π oder $\sqrt{2}$ irrationale Zahlen sind, die sich nicht durch einen Bruch

 © 1990 technicSupport

darstellen lassen und deren Wert sich auch durch noch so viele Stellen nach dem Komma einfach nicht exakt angeben läßt.

Mit etwas Glück trifft man dann im Laufe seines Lebens auf einen Unbelehrbaren, der glatt behauptet, er hätte eine Lösung für die Gleichung

$$x^2 = -1$$

obwohl eigentlich jeder weiß, daß x^2 immer einen positiven Wert ergibt, denn auch wenn x eine negative Zahl ist, ergibt minus mal minus doch immer wieder plus. Der freche Kerl behauptet jedoch ganz dreist, daß die Lösung „i“ laute, und auf unsere Frage, was das wohl sein solle, setzt er noch eins drauf: „i“ sei eben genau die Zahl, deren Quadrat -1 ergibt. Den Namen „i“ (imaginär) habe er ihr gegeben, weil es sie ja eigentlich gar nicht gäbe, sondern sie nur in seiner Vorstellung existiere. Das Pferd beim Schwanz aufgezäumt, nennt man das ja wohl! Doch nun wieder etwas ernster: Diese „künstliche“ Zahl „i“ spielt eine bedeutende Rolle in der modernen Physik, speziell zum Beispiel in der Quantentheorie, aber auch in den anderen modernen Naturwissenschaften; auch unsere Apfelmännchen wären ohne sie nicht denkbar. Es gilt also folgende Definition:

$$i := \sqrt{-1}$$

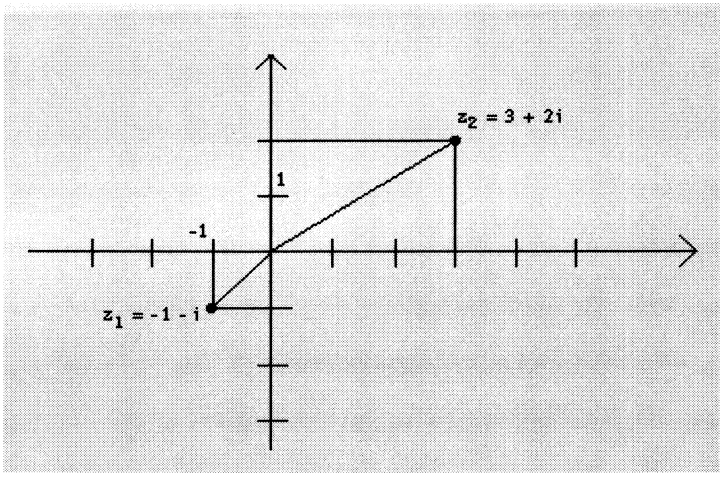
Hat man die Zahl i auf diese Weise definiert, kann man sie mit den reellen Zahlen verknüpfen, zum Beispiel durch Multiplikation und Addition. Man erhält dadurch Zahlen wie:

$$3 \cdot i \quad -5 \cdot i \quad \sqrt{2} \cdot i \quad i + 2 \quad \pi \cdot i - 2 \quad usw.$$

Zahlen, die sich auf diese Weise bilden lassen, werden komplexe Zahlen genannt. Eine solche komplexe Zahl z läßt sich durch die Formel

$$z = 1 \cdot x + i \cdot y$$

darstellen, wobei x und y jeweils reelle Zahlen sind. Eine komplexe Zahl besteht demnach aus zwei verschiedenen Teilen. Die erste Hälfte $(1 \cdot x)$ ist immer eine reelle Zahl, die sich auch auf dem Zahlenstrahl darstellen läßt. Dieser Teil wird als Realteil bezeichnet. Der zweite Teil $(i \cdot y)$ bildet ein Vielfaches der Zahl i. Da i keine reelle



Die Ebene der komplexen Zahlen

Zahl ist, kann weder sie noch ein Vielfaches von ihr auf dem Zahlenstrahl abgebildet werden. Wie also kann eine komplexe Zahl grafisch dargestellt werden? Eine von mehreren Methoden besteht darin, auf einen zweiten Zahlenstrahl zurückzugreifen, der senkrecht auf dem ersten steht. Auf diesem zweiten Strahl stellt man dann nicht die Werte für $i \cdot y$, sondern nur für die reelle Zahl y dar. Da y den Anteil repräsentiert, der mit der imaginären Zahl i multipliziert wird, wird y als Imaginärteil der komplexen Zahl bezeichnet.

Auf diese Weise erhält man ein Koordinatensystem, welches eine Ebene aufspannt, wie es in der Abbildung zu sehen ist. Jeder Punkt dieser Ebene entspricht genau einer komplexen Zahl, deren Realteil sich auf der x-Achse und deren Imaginärteil sich auf der y-Achse ablesen läßt. In der Abbildung finden sich zwei Beispieleinträge für die Zahlen $z_1 = -1 - i$ sowie $z_2 = 3 + 2 \cdot i$.

 © 1990 technicSupport

Ein wichtiger Aspekt im Hinblick auf `MandelVroom` bleibt noch zu erörtern: die „Größe“ einer komplexen Zahl, die von Bedeutung ist, wenn es darum geht, das Verhalten einer Zahlenfolge zu bestimmen. Welche der folgenden Zahlen ist denn nun die größte?

$$z_1 = -1 - i; \quad z_2 = 1 + i; \quad z_3 = 1 - i; \quad z_4 = -3 - 4 \cdot i$$

Diese auf den ersten Blick triviale Frage ist gar nicht so leicht zu beantworten, weil man sich unter der Zahl i nicht besonders viel vorstellen kann. Trägt man aber die entsprechenden Punkte in das Koordinatensystem ein, dann sieht man, daß sich eigentlich nur ein Verfahren zur Größenbestimmung anbietet. Man definiert die Größe (der Mathematiker spricht hier vom *Betrag*) einer komplexen Zahl als Abstand des entsprechenden Punktes vom Mittelpunkt des Koordinatensystems.

Betrachten wir die Strecke vom entsprechenden Punkt zum Mittelpunkt: Sie bildet mit ihren Projektionen auf die x- bzw. y-Achse ein rechtwinkliges Dreieck, so daß sich ihre Länge über den Satz des Pythagoras ausrechnen läßt. Diese Länge ist aber genau der Betrag der komplexen Zahl z, üblicherweise geschrieben als $|z|$. Damit gilt:

$$|z|^2 = x^2 + y^2 \qquad bzw. \qquad |z| = \sqrt{x^2 + y^2}$$

Für die vier oben genannten Zahlen beispielsweise ergeben sich die folgenden Beträge:

$$|z_1| = \sqrt{(-1)^2 + (-1)^2} = \sqrt{1+1} = \sqrt{2}$$

$$|z_2| = \sqrt{(+1)^2 + (+1)^2} = \sqrt{1+1} = \sqrt{2}$$

$$|z_3| = \sqrt{(+1)^2 + (-1)^2} = \sqrt{1+1} = \sqrt{2}$$

$$|z_4| = \sqrt{(-3)^2 + (-4)^2} = \sqrt{9+16} = \sqrt{25} = 5$$

© 1990 technicSupport

Die Mandelbrot-Menge

Nach dieser langen Vorrede können wir uns nun wieder der Zahlenfolge

$$z_{n+1} := z_n^2 + c$$

zuwenden, deren Verhalten wir bereits oben für ein einfaches Beispiel untersucht haben. Entsprechend einfach und nicht sonderlich interessant war das Ergebnis ausgefallen. Jetzt wollen wir diese Folge aber nicht mehr für reelle, sondern für komplexe Zahlen untersuchen. Außerdem legen wir fest, daß der Anfangswert z_0 immer Null ist. Untersuchen wollen wir das Verhalten der Zahlenfolgen diesmal für unterschiedliche Werte der komplexen Zahl c.

Für $c = 0$ ist die Lösung noch ganz einfach: Egal wie groß n wird, z_n bleibt immer Null. Bei anderen Werten für c ist das Verhalten der Zahlenfolge nicht mehr so leicht zu bestimmen. Genau diese Aufgabe ist es, die `MandelVroom` für uns erledigt.

Das nach dem Programmstart geöffnete Fenster besteht ja aus sehr vielen Bildpunkten. Irgendwo innerhalb dieses Fensters liegt auch der Mittelpunkt des besprochenen Koordinatensystems. Aus der Position eines Bildpunktes zu diesem Nullpunkt ermittelt `MandelVroom` den Wert der komplexen Zahl c, die diesem Bildpunkt zugeordnet ist. Dieser Wert wird in die obere Gleichung eingesetzt. Die Zahl z_0 wird gleich Null gesetzt. Aus diesen beiden Werten wird z_1 berechnet, das Ergebnis wird anstelle von z_0 eingesetzt, womit dann z_2 errechnet werden kann.

Dieser Vorgang wird fortgesetzt, bis entschieden werden kann, ob die Beträge der Zahlen z_n bei wachsendem n begrenzt bleiben oder nicht. Bleibt der Betrag für einen Wert c begrenzt, dann verleiht `MandelVroom` dem zugehörigen Bildpunkt die Farbe Schwarz. Diese Berechnung muß für jeden Bildpunkt des Fensters neu durchgeführt werden, was dann auch die Erklärung für die lange Rechenzeit liefert.

Die bizarre, schwarze Figur, die uns `MandelVroom` präsentiert, besteht somit aus der Menge aller komplexen Zahlen c, für welche die

 © 1990 technicSupport

Zahlenfolge

$$z_{n+1} := z_n^2 + c; z_0 = 0$$

begrenzt bleibt. Genau diese Zahlenmenge ist es, die man als MANDELBROT-Menge, ein wenig trivialer und wegen ihrer Form aber auch als Apfelmännchen bezeichnet. Außerhalb dieser Menge liegen logischerweise alle die komplexen Zahlen, für welche die Folge nicht beschränkt bleibt. Für den Betrachter ist in erster Linie der Grenzverlauf zwischen diesen beiden Zahlenmengen interessant, die nichts mehr von dem langweiligen Kreis an sich hat.

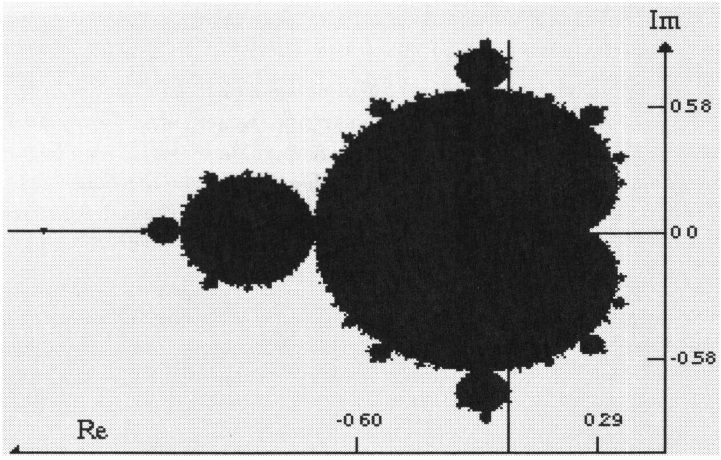
Für eingehendere Betrachtungen sind aber auch die Zahlenwerte c von Interesse, denen die Bildpunkte entsprechen. Sobald in einem Fenster eine Figur auch nur stückweise gezeichnet ist, kann man den Mauszeiger auf eine beliebige Stelle des Fensters führen und die linke Maustaste drücken. In der Titelleiste des Screens erscheint dann eine Information der Art:

```
Height 4  r 0.605803  i  -0.044709
```

Die auf `r` folgende Zahl gibt den Realteil für den Bildpunkt an, auf dem sich der Mauszeiger gerade befindet. Es ist nicht schwer zu erraten daß die Zahl hinter dem `i` den Imaginärteil darstellt. Wandert man mit gedrückter linker Maustaste durch das Fenster, bemerkt man, daß die sich die Zahlen nicht kontinuierlich, sondern sprunghaft ändern. So wird man es in der Regel auch nicht schaffen, den genauen Nullpunkt zu treffen. Hier macht sich die begrenzte Auflösung des Screens bemerkbar, die — neben anderen Faktoren — auch dafür verantwortlich ist, daß die Grenzlinie nicht so fein dargestellt ist, wie es wünschenswert wäre. Aber **MandelVroom** besitzt ja eine Zoom-Funktion, die wir später noch einmal benutzen werden, um dieses Manko zu beheben.

Was man auf Grund der niedrigen Auflösung nicht erkennen kann: Jeder schwarze Punkt ist über weitere schwarze Punkte mit den anderen Punkten dieser Menge verbunden. Bisweilen sind die Verbindungen zwischen den Punkten allerdings so fein, daß sie erst beim Vergrößern eines Ausschnittes sichtbar werden. Eine solche Menge wollen wir im folgenden als *konnex* bezeichnen.

© 1990 technicSupport



Die berühmte Mandelbrotmenge

Feldlinien

Einen großen Teil ihres Reizes verdanken diese Grafiken neben dem Grenzverlauf der Mandelbrot-Menge den farblich abgestuften Flächen um diese Menge herum. Den physikalisch vorbelasteten Leser werden die Grenzen zwischen diesen Flächen vielleicht schon an die von der Elektrostatik oder der Gravitation bekannten Feldlinien erinnert haben. Und in der Tat besteht eine Analogie zwischen diesen Feldlinien und den Verläufen außerhalb der Mandelbrot-Menge. Erinnert man sich daran, daß unsere Zahlenfolge für die Punkte der Mandelbrot-Menge beschränkt bleibt, dann kann man sich mit der Vorstellung behelfen, daß diese geschlossene Menge eine Art Anziehungskraft auf alle außerhalb gelegenen Punkte ausübt, die das Ansteigen der Zahlenfolge für den jeweiligen Punkt bremst. Vereinfacht ausgedrückt, hängt die Stärke der Anziehungskraft davon ab, wie dicht der Punkt an der Mandelbrot-Menge liegt. Es spielt je-

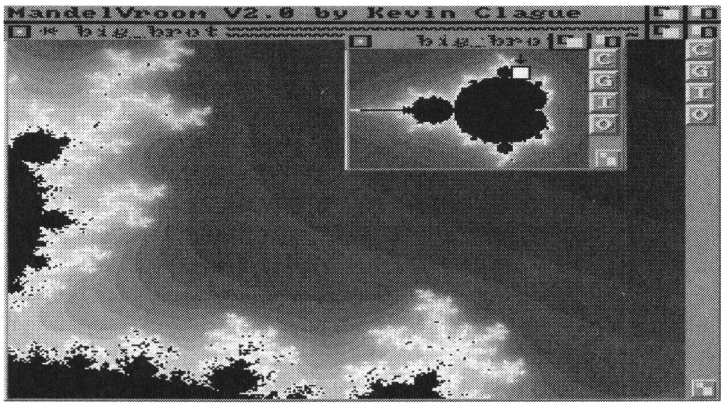
 © 1990 technicSupport

doch auch eine Rolle, ob der Punkt in einer Einkerbung oder etwa dicht an einem antennenartigen Auswuchs liegt.

In der Regel wächst die Zahlenfolge daher umso schneller an, je weiter der Punkt von der Mandelbrot-Menge entfernt ist. Wie schnell eine Zahlenfolge wächst, ermittelt `MandelVroom`, indem es den jeweiligen Betrag einer Zahl z_n mit einem festgesetzten Grenzwert vergleicht. Ist dieser Grenzwert bei einem bestimmten n erreicht bzw. überschritten, wird die Berechnung abgebrochen und n als Maß für die Wachstumsgeschwindigkeit der Zahlenfolge gewertet. Eine Folge, bei der bereits $|z_2|$ den Grenzwert erreicht, wächst schneller als eine, die den Grenzwert erst mit $|z_9|$ erreicht. Bei welchem Wert n die Zahlenfolge für einen bestimmten Punkt den Grenzwert erreicht oder überschreitet, läßt sich ebenfalls nach dem Anklicken eines Punktes aus der Titelleiste ablesen. Dort steht der entsprechende Zahlenwert hinter der Bezeichnung *Height* (Höhe). Diese Bezeichnung wurde gewählt weil sich auch ein dreidimensionales Modell der Mandelbrot-Menge denken ließe, in welchem n dem Punkt als Höhe auf einer z-Achse zugeordnet werden könnte. Oft bezeichnet man diese Größe aber als Feldstärke.

Da ein Bildschirm aber nur zweidimensional ist, drückt `MandelVroom` diesen Wert durch eine entsprechende Farbgebung aus. Klicken Sie sich einmal mit der linken Maustaste durch die verschiedenen Farbbereiche. Sie werden feststellen, daß die angezeigte Feldstärke (Höhe) von außen nach innen zunimmt. Einige Farbbereiche enthalten nur Punkte derselben Feldstärke, andere enthalten Punkte, deren Feldstärke sich innerhalb eines bestimmten Bereiches bewegt. Würde man jedem Feldstärkewert eine eigene Farbe zuordnen, müßte man schon im HAM-Modus arbeiten (den unterstützt das Programm aber nicht), auch würde in vielen Fällen die Aussagekraft des Bildes darunter leiden.

Was geschieht aber bei solchen Punkten, für die die Zahlenfolge den Grenzwert niemals erreicht? Muß Amiga rechnen, bis er schwarz wird? Müssen wir ewig auf ein Bild warten? Um solchen Enttäuschungen vorzubeugen, legt man einfach einen Maximalwert für die Zahl n fest. `MandelVroom` ist so voreingestellt, daß für die Berechnung einer Zahlenfolge höchstens 128 Durchgänge (Iterationen) er-



Ein Stück des Grenzbereiches wird vergrößert

laubt sind. Ist für z_{128} der Grenzwert noch nicht erreicht, dann wird der Punkt der Mandelbrot-Menge zugeordnet. Aus diesem Grunde besitzen alle schwarzen Punkte eine Feldstärke von 128.

Jetzt wird auch verständlich, warum das Vergrößern eines Ausschnittes länger dauert, wenn er viele schwarze Punkte enthält. Für jeden dieser Punkte muß die Gleichung 128mal berechnet werden, bei Punkten außerhalb dieser Menge sind nur soviele Durchgänge nötig, bis der Grenzwert erreicht ist. Wenn Sie jetzt noch einmal auf das [G] -Gadget eines Fensters klicken und scharf hinsehen, werden Sie deutlich erkennen, daß der Aufbau der mittleren Zeilen, die am meisten schwarze Punkte enthalten, wesentlich mehr Zeit in Anspruch nimmt als die Berechnung der oberen und unteren Randzeilen.

Die Erforschung der Mandelbrotmenge

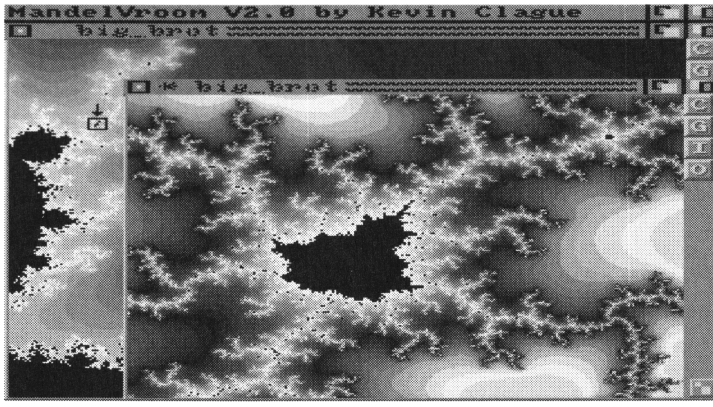
Mit diesem Grundwissen gerüstet, können wir das Programm noch einmal neu starten. Wie bereits erwähnt, kann das erste Bild —

 © 1990 technicSupport

wie übrigens jedes andere Bild — den Grenzverlauf zwischen der Mandelbrotmenge und den übrigen Bildpunkten nur unvollkommen abbilden. In Wirklichkeit ist dieser Grenzverlauf viel komplizierter. Aus jeder Fläche, die in der jetzigen Auflösung noch Ähnlichkeiten mit einem Kreis aufweist, wachsen weitere Ausbuchtungen, aus diesen Ausbuchtungen antennenartige Gebilde usw., die in der jetzigen Auflösung und Bildgröße nicht erkennbar sind. Man kann dieses Bild in etwa mit der Satellitenaufnahme von einem Kontinent vergleichen, zum Beispiel von Australien, das in etwa die Form einer Bohne zeigen würde. Mit einer höheren Vergrößerung könnte man bei Melbourne eine Einbuchtung und das vorgelagerte Tasmanien erkennen. Bei noch stärkeren Vergößerungen würden weitere, kleinere Buchten deutlich, schließlich auch die gerippte Struktur des Sandstrandes, unter einem Mikroskop am Ende auch die feine Struktur der Sandkörner.

Auf vergleichbare Weise können wir die Zoom-Funktion von `MandelVroom` nutzen, um bestimmte Ausschnitte des Grenzabschnittes zu vergrößern. Dazu ist, wie oben beschrieben, ein neues Mandelbrot-Fenster zu öffnen. Dieses Fenster kann man auf die volle Bildschirmgröße bringen, es in den Hintergrund schalten, und auf das [I] -Gadget des noch leeren Fensters klicken. Anschließend wird ein Ausschnitt aus dem bereits vorhandenen Fenster herausgeschnitten und durch Betätigen des [G] -Gadgets vergrößert. Das Ergebnis läßt sich aus der Abbildung ablesen. Mit der `Autocontour`-Funktion (s. u.) kann die Darstellung der Potentialverläufe verbessert werden.

Aus diesem Bild kann nun ebenfalls ein bestimmter Bereich herausgegriffen und vergrößert werden, wie es die nachstehende Abbildung zeigt. Aus dem unscheinbaren „Punkt“ entsteht dabei wieder eine Art Apfelmännchen: Es ist wirklich ein solches, nur bewegen wir uns bereits in einem so feinen Bereich, daß Rechenungenauigkeiten die Qualität der Darstellung vermindern. Durch Drücken der linken Maustaste und Hin- und Herbewegen der Maus läßt sich an der Koordinatenanzeige in der Titelleiste ablesen, daß unser Bild nur noch einem vergleichsweise kleinen Bereich der komplexen Zahlenebene entspricht. Aus den kleinen schwarzen Punkten läßt sich auch erahnen, daß diese kleinen Äpfelmännchen mit dem großen durch



Nochmaliges Vergrößern fördert weitere Details zutage

„hauchdünne“ Linien verbunden sind. Auf noch mehr Apfelmännchen stößt man übrigens auch, wenn man noch weiter vergrößert. Man könnte dieses Spiel ohne Ende fortsetzen, die Randstrukturen würden sich lediglich immer feiner verästeln, und immer wieder würde man Apfelmännchen finden, die in unbegrenzter Zahl vorhanden sind: Dieses Phänomen, das auch als Selbstähnlichkeit bezeichnet wird, ist nicht nur für den unbefangenen Betrachter, sondern auch für den ernsthaften Forscher und Mathematiker ein besonders faszinierender Aspekt dieser Materie.

Julia-Mengen

Bei unseren ersten Schritten mit `MandelVroom` haben wir uns stets mit der Mandelbrot-Menge beschäftigt. Es war immer ein und dieselbe Zahlenmenge, die wir lediglich mehr oder weniger detailliert betrachtet haben. Diese Menge bestand aus allen komplexen Zahlen c, für welche die Folgen

$$z_{n+1} := z_n^2 + c; z_0 = 0$$

 © 1990 technicSupport

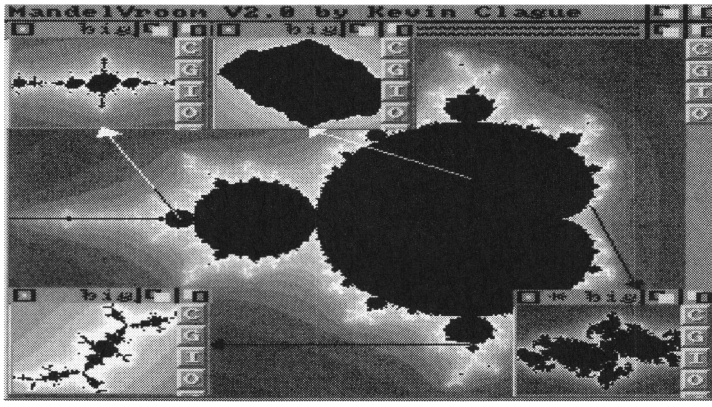
beschränkt blieben. Aus genau derselben Formel lassen sich aber noch andere Zahlenmengen ableiten, die ebenfalls zu ansehnlichen Computergrafiken führen und für Mathematik und Wissenschaft von nicht minderem Interesse sind als die Mandelbrot-Menge: die Julia-Mengen.

Eine Julia-Menge ist definiert als die Menge aller Zahlen z, für welche die Folge

$$z_{n+1} := z_n^2 + c; z_0, c \in C$$

beschränkt bleibt. Der entscheidende Unterschied zur Mandelbrot-Menge liegt darin, daß das Verhalten der Zahlenwerte nun für die komplexen Zahlen z_0 untersucht wird. Die Koordinaten, die `MandelVroom` anzeigt, beziehen sich jetzt also auf diese Zahlen und nicht mehr auf c. Ein weiterer Unterschied: Für c darf jede beliebige komplexe Zahl eingesetzt werden, und jeder Zahl c ist damit auch eine bestimmte Julia-Menge zugeordnet. Das bedeutet, daß es nur eine Mandelbrot-Menge gibt, aber unendlich viele Julia-Mengen. Weil es diese Zahl c ist, durch die eine Julia-Menge bestimmt ist, wird sie gewöhnlich als *Seed* bezeichnet, was soviel bedeutet wie Saatkorn oder Keim.

Dieses Korn pickt sich `MandelVroom` aus einem bereits bestehenden Mandelbrot-Bild heraus; es ist nicht möglich, den Wert in Form einer Zahl einzugeben. Zuvor muß allerdings im Menü `Project/New` ein neues Fenster geöffnet werden, wobei diesmal nicht `Mand` sondern `Julia` anzuwählen ist. Ein solches Julia-Fenster besitzt gegenüber den Mandelbrot-Fenstern ein zusätzliches Gadget: [J] . Nach dem Anklicken dieses Feldes kann der Mauszeiger auf einen beliebigen Punkt eines Mandelbrot-Bildes geführt werden. Drückt man dort die linke Maustaste, so wird die diesem Punkt zugeordnete Zahl c als *Seed* für das Julia-Bild gesetzt. Die Berechnung dieses Bildes wird auf gewohnte Weise über [G] in Gang gesetzt. Ist das Bild fertig gerechnet, können Ausschnittsvergrößerungen angefertigt werden, wozu ein neues Julia-Fenster geöffnet und der Ausschnitt jetzt wieder über das [I] -Gadget bestimmt wird. Das [J] -Gadget spielt jetzt keine Rolle mehr, es wird nur benötigt, um anfangs eine ganz spezielle Julia-Menge festzulegen.

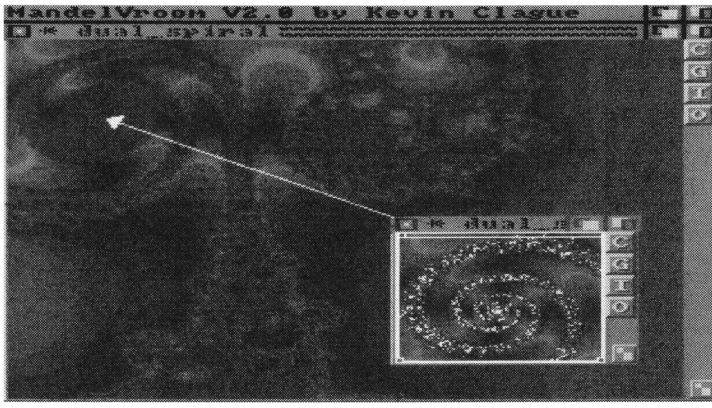


Julia-Mengen für verschiedene Seeds

Julia-Mengen lassen sich grob in zwei Gruppen unterteilen: in konnexe Mengen und in Mengen, bei denen nicht alle Punkte miteinander verbunden sind. Zu welcher dieser Gruppen eine Julia-Menge gehört, hängt davon ab, welcher Punkt eines Mandelbrot-Bildes beim Festlegen des *Seeds* angeklickt wird. Ist es ein schwarzer Punkt (d. h. ein Punkt der Mandelbrot-Menge), dann bleibt die Julia Menge konnex, ansonsten tritt das krasse Gegenteil ein. Dies ist dann auch der elementare Zusammenhang zwischen der Mandelbrot- und den Julia-Mengen. Für die Julia-Mengen in Abhängigkeit von der Zahl c bedeutet das, daß ihr Charakter während des Übergangs von den Punkten c im Inneren der Mandelbrot-Menge zu den außerhalb gelegenen einer dramatischen Wandlung unterworfen wird. Die interessantesten Julia-Mengen wird man demnach finden, wenn man ein Mandelbrot-Bild vergrößert und mit verschiedenen *Seeds* im Grenzbereich der schwarzen Fläche experimentiert.

Jetzt wird es aber Zeit, sich den übrigen Funktionen von `MandelVroom` zuzuwenden. Da sind zunächst die beiden bislang unerwähn-

 © 1990 technicSupport



Auch das umgekehrte Zoomen ist möglich

ten Gagdets [C] und [O]. Durch Anklicken des [C]-Gadgets, wird das jeweilige Fenster zum aktuellen erklärt, was durch ein Sternchen in der Titelleiste des Fensters angezeigt wird. Alle folgenden Operationen wie Farbänderungen, Konturieren usw. wirken dann nur auf dieses aktuelle Fenster.

Das [O]-Gadget besitzt die umgekehrte Funktion wie [I]. Nach Anklicken dieses Gadgets erscheint der bekannte, fensterähnliche Rahmen, mit dem sich ein Bildausschnitt festlegen läßt. Der Rahmen bestimmt diesmal aber die Größe, auf die das gesamte Bild *verkleinert* werden soll, seine Position legt fest, an welcher Stelle des neuen Bildes der verkleinerte Ausschnitt erscheinen soll. Nach dem Anklicken von [G] wird das neue Bild berechnet, welches dann auch die Umgebung des alten Bildes zeigt. Das Fenster kann vor dem Rechenvorgang auch vergrößert werden, ein Beispiel zeigt die obere Abbildung.

© 1990 technicSupport

Das Project-Menü

`Current` Das zuvor angeklickte Fenster wird aktiviert (wie [C]).

`Load` Lädt ein sogenanntes Projekt. Ein Projekt ist gleichzusetzen mit dem aktiven Fenster und den für dieses Fenster getroffenen Einstellungen. Ein Projekt enthält ebenfalls alle Informationen über die Feldstärken, d. h. über die Konturierung, so daß das zugeordnete Bild nicht neu berechnet werden muß und recht schnell aufgebaut werden kann. Natürlich kann mit diesem Bild dann auch weitergearbeitet werden.

`Save` Speichert ein Projekt unter frei wählbarem Namen. Wegen der langen Rechenzeiten sollte von dieser Funktion oft Gebrauch gemacht werden: Es ist empfehlenswert, jedes Bild, mit dem man zufrieden ist, als Projekt abzuspeichern, damit man über die `Load`-Funktion schnell wieder darauf zurückgreifen kann, ohne den Rechenvorgang wiederholen zu müssen.

`Close` Schließt das aktuelle Fenster.

`SaveILBM` Speichert den gesamten Bildschirm als IFF-ILBM Bild. Diese Bilder können mit Malprogrammen weiterbearbeitet werden. Ein Einladen in `MandelVroom` ist nicht mehr möglich.

`Help` Stellt eine umfangreiche Hilfsfunktion zur Verfügung. Nach dem Aufrufen dieses Menüpunktes ändert sich der Mauszeiger in „With“. Wird daraufhin ein Gadget angeklickt oder ein Menü aufgerufen, so wird nicht die gewohnte Funktion ausgelöst, sondern ein Fenster geöffnet, in dem ein Hilfstext über die entsprechende Funktion informiert. Durch zweimaliges Anwählen von `Help` erscheint sogar ein Hilfstext für die `Help`-Funktion.

`Cancel` Bricht eine Operation ab, die einen weiteren Parameter über die Maus erfordert (z. B. [I]).

`Quit` Das Programm wird nach einer Sicherheitsabfrage verlassen.

 © 1990 technicSupport

Das Display-Menü

`Colors` Öffnet eine Farbpalette zum Ändern der Bildschirmfarben.

`Cycling` Die Bildschirmfarben werden in regelmäßigen Abständen zyklisch miteinander vertauscht. Ein kleiner optischer Zusatzeffekt.

`Contours` Dient zum Konturieren eines Bildes per Hand (s.u.).

`Autocontour` Hier stehen zwei Möglichkeiten zum automatischen Konturieren eines Bildes zur Verfügung.

`Histogram` Zeigt in einem Fenster die prozentuale Verteilung der Feldstärken (Höhen) an. Diese Verteilung kann Aufschlüsse für die Wahl der Feldstärkenbereiche geben, die beim Konturieren derselben Farbe zugeordnet werden sollen.

`Borderless` Das aktive Fenster wird ohne Rand, also auch ohne Gadgets, dargestellt. Anstelle der Gadgets sind dann die äquivalenten Menüfunktionen zu benutzen.

`Num Colors` Legt die Zahl der Farben fest, die auf dem Bildschirm dargestellt werden. Der Maximalwert hängt vom gewählten Grafik-Modus ab.

`View Modes` Durch Anwahl und Kombination der Untermenüs `Hi Res`, `Interlace` und `HalfBrite` werden Bildschirmauflösung und maximale Farbzahl (HalfBrite) bestimmt. Die Kombination `Halfbrite` mit `Interlace` ist nicht möglich. An den PAL-Bildschirm paßt sich das Programm von selbst an.

`Screen Size` Erlaubt die Wahl zwischen Standard-Bildschirm und Größe des Workbench-Screens, was nur dann von Bedeutung ist, wenn die Workbench im Overscan-Modus betrieben wird und dieser Modus von `MandelVroom` genutzt werden soll.

© 1990 technicSupport

Das Calculate-Menü

Generate Die Funktionen **Start** und **Stop** entsprechen dem [G] -Gadget. Mit **Resume** läßt sich ein unterbrochener Rechenvorgang fortsetzen, ohne daß die bereits errechneten Zeilen neu berechnet werden.

Zoom Die Funktionen **In**, **Out** und **Julia** entsprechen den Gadgets [I] , [O] und [J] . Mit **Close** läßt sich der fensterähnliche Rahmen schließen.

Pan Nach Anwahl des **Start**-Untermenüs kann ein Bild in seinem Fenster verschoben werden. Es entstehen freie Randbereiche, die mit **Generate** berechnet werden können. Der sichtbar gebliebene Teil des Bildes wird nicht neu berechnet.

Magnifier Hier läßt sich die Lupenfunktion beim Vergrößern eines Ausschnittes unterdrücken. Das Scrollen des Ausschnittes verläuft dann ruckfreier.

Math Mode Bestimmt das Rechenverfahren, von dem die Genauigkeit der Berechnung, aber auch die Geschwindigkeit abhängt. Der normale Amiga-Anwender hat die Wahl zwischen **Int68000** (Ganzzahlen, schnell aber ungenau), **FFP** (Fließkommazahlen, genauer, aber langsamer) und **IEEE** (64-Bit Fließkommazahlen, sehr genau, aber extrem langsam). Die schnelleren Prozessoren einer 68020/881(2)-Karte können mit den Einstellungen **68020** und **68881** voll ausgenutzt werden.

Max Iteration
Es erscheint ein Gadget, in dem die maximale Zahl n der Rechendurchgänge für einen Punkt und damit auch die maximale Feldstärke festgelegt werden kann. Voreingestellt ist ein Wert von 128. Höhere Werte verbessern die Genauigkeit der Darstellung im Grenzbereich, erhöhen die Rechendauer aber beträchtlich, besonders bei Bildern, die viele schwarze Punkte enthalten.

Statistics Ein Fenster zeigt einige Parameter des aktuellen Fensters.

 © 1990 technicSupport

Das Special-Menü

`Preset` Erlaubt das Laden von neun vordefinierten Mandelbrot- bzw. Julia-Bildern, die gerade dem Neuling als Ausgangspunkt für weitere Forschungen dienen.

`Orbit` Dieses Menü öffnet ein weiteres Fenster, in dem beim Anklicken eines Punktes aus einem anderen Fenster mehrere scheinbar willkürlich verteilte Punkte gezeigt werden. Jeder Punkt entspricht dabei genau einer Zahl z_n der für den angeklickten Punkt berechneten Zahlenfolge. Der Abstand eines „Orbit"-Punktes zum Mittelpunkt des Orbit-Fensters repräsentiert den Betrag $|z_n|$ der komplexen Zahl, dessen Verhalten für wachsendes n ja die entscheidende Rolle spielt. Klickt man in die Nähe des Apfelmännchen-Mittelpunktes, sind auch die Punkte im Orbit-Fenster sehr dicht um den Mittelpunkt herum verteilt, weil die Zahlenfolge für Punkte aus diesem Bereich sehr langsam wächst. Daß man nicht die erwartete Zahl von 128 Punkten sieht (für jeden schwarzen Punkt wird doch eigentlich 128 mal gerechnet), hat seinen Grund unter anderem (vgl. `Max Orbit`) darin, daß sie so dicht zusammenliegen, daß sie nicht mehr aufgelöst werden können. Bewegt man sich nun bei gedrückter linker Maustaste vom Mittelpunkt des Apfelmännchens weg, so entfernen sich auch die Punkte im Orbit-Fenster immer weiter von dessen Mittelpunkt, weil die Zahlenfolgen immer schneller wachsen.

`Orbit Mode` Legt das Rechenverfahren für die Orbit-Darstellung fest.

`Max Orbit` Für das Orbit-Fenster wird jede Zahlenfolge neu berechnet. Wie oft, d. h. bis zu welchem n, wird über diesen Menüpunkt eingestellt. Bei Voreinstellung auf 32 können im Orbit-Fenster folglich auch höchstens 32 Punkte erscheinen. Nach Ändern dieses Wertes muß das `Orbit`-Menü übrigens erneut angewählt werden.

Farbeffekte und Konturieren

Die ästhetische Wirkung eines Mandelbrot- oder Julia-Bildes hängt entscheidend von einer geschickten Farbwahl und gut abgestuften Konturen ab. Besonders die letzte Aufgabe ist alles andere als einfach und erfordert viel Erfahrung; man wird es sich daher oft bequem machen und auf das `Autocontour`-Menü zurückgreifen.

Die Auswahl der Bildschirmfarben ist noch recht einfach: Über das `Colors`-Menü wird eine Farbpalette geöffnet, in der jede Farbe mit einem Mausklick zur Bearbeitung ausgewählt werden kann. Die Farbe wird durch Verschieben der drei RGB-Schieberegler auf den gewünschten Farbton eingestellt. Durch Anklicken einer Farbe, Anklicken des [Copy] -Gadgets sowie Anklicken einer anderen Farbe, kann die erste Farbe kopiert werden. Auf dieselbe Weise können mit [Exchg] zwei Farben miteinander vertauscht werden. Nach Wahl einer Farbe kann über [Spread] und Anklicken einer zweiten Farbe ein abgestufter Farbübergang zwischen beiden Werten erzielt werden.

Color Cycling ist über das `Cycling`-Menü zugänglich, setzt aber voraus, daß die Farbpalette geöffnet ist. Man arbeitet jetzt quasi über Kreuz mit zwei Fenstern, woran man sich ein wenig gewöhnen muß. Zunächst wählt man eine Farbe aus der Palette an, betätigt [Range] im `Cycling`-Fenster und klickt anschließend eine weitere Farbe aus der Palette an. Die beiden Farben begrenzen den Bereich, in dem die Vertauschung stattfindet. Nach Anklicken des [Off/On] -Gadgets ist das Color Cycling bereits aktiviert, der Effekt wird aber erst sichtbar, wenn mit dem Schieberegler die Durchlaufgeschwindigkeit erhöht wird. Die Richtung wird mit dem Pfeil in der unteren rechten Ecke bestimmt. Über die Gadgets [c1] , [c2] , usw. sollen sich verschiedene Bereiche programmieren lassen, diese Funktion versagte jedoch ihren Dienst.

Die heikelste Aufgabe ist das Konturieren eines Bildes, das wir am Beispiel der ursprünglichen Mandelbrot-Menge (big_brot) betrachten wollen. Auch hier muß die Farbpalette geöffnet sein, wenn man Farbtöne ändern möchte. Das `Contour`-Menü stellt ein Fenster bereit, in dem zunächst eine Reihe von Schiebereglern auffällt, über

 © 1990 technicSupport

denen sich jeweils ein kleines Quadrat befindet. Dieses Quadrat zeigt die Farbe der jeweiligen Kontur, dient aber gleichzeitig dazu, eine Kontur für die Bearbeitung anzuwählen.

Klickt man das schwarze Quadrat ganz links an, werden in der Titelleiste des Fensters drei Werte für diese Kontur angezeigt: `C` steht für die Nummer der Kontur, die Numerierung verläuft von links nach rechts, und zwar von eins bis 255. Soviele Schieberegler sind zwar nicht zu sehen, man kann jedoch mit dem Mauszeiger den kleinen Rahmen greifen, in dem der Farbverlauf angezeigt wird, und ihn nach rechts ziehen, womit man Zugriff auf die weiteren Konturen erhält. Verwendet man mehr Konturen, als Farben vorhanden sind, müssen Farben mehrfach benutzt werden. Zurück zur Titelleiste: Der Wert `P` steht für die Nummer der Farbe und der wichtigste Wert, `H`, steht für den Feldstärken- bzw. Höhenbereich, der in der entprechenden Farbe dargestellt wird. Schwarz sind daher nur die Punkte, die eine Höhe von genau 128 besitzen. Klickt man auf das Quadrat rechts daneben (Kontur 2), stellt man fest, daß diese Farbe allen Punkten mit einer Höhe von 127–29 zugeordnet wird. Die dritte Farbe steht für alle Punkte mit einer Höhe von 28–19 usw. Es besteht also kein linearer Zusammenhang.

Die Höhenbereiche, denen eine Kontur zugeordnet ist, lassen sich durch Greifen und Verschieben des Schiebereglers mit der linken Maustaste ändern. Kleinere Schritte können durch Mausklicke ober- bzw. unterhalb der Knöpfe erzielt werden. Mit dem Schieberegler ganz rechts kann auf Wunsch die Übersetzung der Regler geändert werden. Wir können einmal die zweite Kontur anklicken und den Schieberegler ein wenig nach unten verschieben. Nehmen wir an, daß die zugeordnete Höhe nach dieser Änderung den Bereich 127–26 umfaßt, dann wird damit automatisch auch der Bereich der nächsten Kontur beeinflußt, die jetzt mit dem nächstniedrigen Wert 25 beginnt. Die Konturierung sollte daher immer auf der linken Seite begonnen werden.

Das bedeutet auch, daß ein Regler niemals dieselbe Höhe erreichen darf wie ein links von ihm befindlicher. Ein Regler, der auf einen solchen falschen Wert gezogen wurde, wird in einer dunkleren Farbe dargestellt, die zugehörige Farbe bleibt unberücksichtigt. Hat man

die gewünschten Änderungen vorgenommen, genügt ein Klick auf [Paint] , und das aktuelle Fenster wird mit den neuen Konturen versehen. Nun muß man aber nicht unbedingt jeden Schieberegler per Hand setzen. Es gibt auch die Möglichkeit, zwei weiter auseinanderliegende Regler einzustellen, eine dieser Konturen zu aktivieren, das [Smooth] -Gadget und anschließend die andere Kontur anzuklicken. Zwischen diesen beiden Konturen wird dann für eine lineare Verteilung der übrigen Bereiche gesorgt. Diese Funktion ist bei kleineren Bereichen sehr praktisch, über größere Bereiche ist eine lineare Verteilung aber nicht ratsam, da das Wachstum unserer Zahlenfolgen auch nicht linear erfolgt. Komplett linear konturierte Bilder sehen in der Regel fürchterlich aus.

Möchte man einer Kontur eine andere Farbe verleihen, klickt man diese Kontur an, klickt auf [Set] und anschließend auf eine Farbe aus der Palette (nicht aus dem Kontur-Fenster). Für die Veränderung ganzer Farbfolgen gibt es das [Copy] -Gadget, mit dem sich eine Farbfolge sowohl aus der Farbpalette als auch aus dem Kontur-Fenster in einen Zwischenspeicher kopieren läßt. Man klickt zum Beispiel auf eine Farbe der Palette, auf [Copy] (im Kontur-Fenster!) und auf eine zweite Farbe der Palette. Analog verläuft der Vorgang für die Farben im Kontur-Fenster. Ein kleiner Hinweis zur Farbauswahl: Es ist kein Zufall, daß auf die schwarze Farbe sofort eine sehr helle folgt. Dieser Kontrast hebt den Grenzverlauf besonders deutlich hervor. Es ist also durchaus nicht so, daß ein gleichmäßiger Farbverlauf von hell nach dunkel die besten Resultate liefert.

Im Zwischenspeicher befindet sich nach einer solchen Aktion eine ganz bestimmte Anzahl von Farben. Werden diese Farben an einen anderen Bereich kopiert, dann kann es vorkommen, daß soviele Farben gar nicht in diesen Bereich passen. Dann wird der Bereich soweit gefüllt, wie es möglich ist. Passen mehr Farben in den Zielbereich, als im Zwischenspeicher vorhanden sind, dann wird das Auffüllen so oft wiederholt, bis der Zielbereich gefüllt ist. Der Anfangspunkt des Zielbereiches wird durch Anklicken einer Kontur bestimmt, danach wird [Paste] betätigt. Die nächste Kontur, die angeklickt wird, bestimmt das Ende des Zielbereiches und veranlaßt das Auffüllen mit den Farbwerten aus dem Zwischenspeicher. Sowohl beim Ko-

 © 1990 technicSupport

pieren von Farben in den Speicher als auch beim Auffüllen spielt die Reihenfolge, in der Anfangs- und Endpunkt festgelegt werden eine Rolle für den Farbverlauf. Das klingt alles ziemlich kompliziert, und es dauert auch geraume Zeit, bis man diese Verfahren sicher beherrscht. Daher ist es ein wenig schade, daß es hier keine Undo-Funktion gibt, die bei unbeabsichtigten Effekten den alten Zustand wieder herstellen könnte. Hier hilft nur Üben, Üben und nochmals Üben. Aber es gibt ja auch noch die `Autocontour`-Funktion. Ebenfalls hilfreich ist es übrigens, verschiedene Presets zu laden und ihre unterschiedlichen Konturen zu studieren.

Fast hätten wir drei Gadgets vergessen! Mit [Cut] läßt sich ein auf die gewohnte Weise festzulegender Farbbereich herausschneiden. [Last] bewirkt dasselbe, als ob man die Kontur 255 anklicken würde. Die Zahl, die man in das `Mod`-Gadget einträgt, beeinflußt die Funktionen [Cut], [Copy] und [Paste]. Steht in diesem Feld zum Beispiel eine 2, dann wird nur jede zweite Farbe herausgeschnitten, kopiert, oder eingefügt. Mit Hilfe dieses Gadgets können daher verschiedene Farbfolgen überlagert, bzw. vermischt werden.

Fazit

`MandelVroom` ist ein Programm, das den an dieser Materie interessierten Anwender stunden- und tagelang an seinen Amiga fesseln wird. So mancher Amiga wird fortan auch in der Nacht damit beschäftigt sein, Bilder in besonders hoher Auflösung oder mit hoher Iterationszahl und im IEEE-Modus zu berechnen.

Über Mandelbrot- und Julia-Mengen ließe sich noch viel sagen, z. B. über die Parallelen zu wissenschaftlichen Theorien über Phasenübergänge oder Bevölkerungswachstum, das kann aber nicht Thema eines Public Domain Buches sein. Interessierte Leser möchten wir daher auf das Buch *The Beauty of Fractals* (H. O. Peitgen, P. H. Richter; 1986, Berlin • Heidelberg • New York • Tokyo) verweisen. Auch die Frage nach einem ästhetischen oder gar künstlerischen Wert der erzeugten Bilder kann und soll in diesem Buch nicht diskutiert werden.

PROBLEME

Wenn man blitzschnell und mehrfach hintereinander die Bildschirmauflösung wechselt — eine nicht sehr praxisnahe Handlungsweise, eher eine Provokation von Seiten des Testers — kann es passieren, daß sich das Programm aufhängt.

HERKUNFT

Das Programm `MandelVroom` belegt in der hier besprochenen Version 2.0 die gesamte Fishdisk 215. Der Quellcode in C befindet sich auf der Fishdisk 214. Der Autor

```
Kevin Clague
2691 Bon Bon Drive
San Jose, CA 95148
(408)-270-0673
USENET: kevin@amdahl.uts.amdahl.com
```

hat alle Rechte an seinem Programm reserviert und freut sich über Kommentare aller Art. Eine Share-Gebühr wird nicht verlangt. Dies ist umso erstaunlicher, weil man sich ohnehin fragt, warum ein so gutes und komplexes Programm nicht auf dem kommerziellen Weg vertrieben wird. In Anbetracht manch anderer teurer Amiga-Programme gibt es hierauf wohl nur eine Antwort: `MandelVroom` ist einfach zu gut gelungen!

VERWEISE

`MandFXP` (Faug 29), `IMandelVroom` (Fishdisk 90), `Julia` (Panorama 31a), `Mandel` (Fishdisk 318), `MSet` (RPD 100), `Chaos-Kit` (RPD 100), `Planet` (RPD 100), `QMan` (Fishdisk 130), `MandelZoom` (RPD 93)

© 1990 technicSupport

MaxWindow

Bringt CLI-Fenster auf PAL-Größe

AUFRUF

Das Programm wird aus dem CLI durch die Eingabe

`MaxWindow`

gestartet.

ZWECK

Das Programm vergrößert das CLI- oder Shell-Fenster, aus dem heraus es aufgerufen wurde, auf die volle Größe des PAL-Bildschirmes.

BESCHREIBUNG

Das Standard CLI- bzw. Shell-Fenster ist für fast alle Vorhaben zu klein. Nahezu immer muß der User erst zur Maus greifen, um das Fenster zu vergrößern. Mit `MaxWindow` geht es einfacher. Nachdem das Programm mit dem Befehl

```
copy A.C.S.#133:maxwindow c:
```

in das `C:`-Verzeichnis der Arbeits-Workbench kopiert worden ist, genügt die Eingabe des Programmnamens, um das Fenster zu vergrößern. Wer mit der Workbench 1.3 arbeitet, kann mit Hilfe eines Editors den Programmaufruf in die Dateien `S:Shell-Startup` und `S:Cli-Startup` einfügen. Da diese Dateien bei jedem Starten eines CLI's oder einer Shell abgearbeitet werden, nehmen die Fenster dann automatisch die maximale Größe an.

HERKUNFT

`MaxWindow` findet man auf der ACS-Disk 133.

Moose

Das Ikea-Imperium schlägt zurück

AUFRUF

Der Aufruf erfolgt über die mitgelieferten Icons von der Workbench aus. Das Starten aus dem CLI ist weiter unten beschrieben.

ZWECK

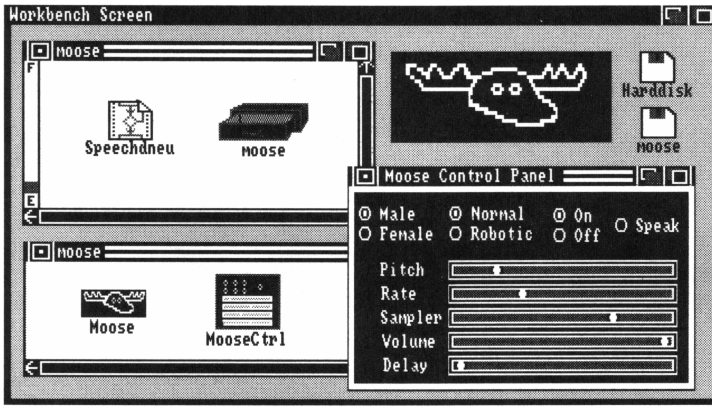
`Moose` bereichert Amiga um einen weiteren sinnlosen, aber unterhaltsamen Hintergrundprozeß.

BESCHREIBUNG

Falls der geneigte Leser in `Moose` ein Lagerverwaltungsprogramm für Möbelgroßhändler oder gar ein interskandinavisches Weltraumballerspielchen — mit 3D-Wasserpistolen und Atomwaffeln vielleicht? — vermutet, muß er sich auf eine herbe Enttäuschung gefaßt machen. Ist er doch auf eine grobe wie auch brilliante Lesertäuschung hereingefallen — frei nach `Eckhard Henscheid`, der einen Band seiner „Trilogie des laufenden Schwachsinns" mit „grober, feister, glänzender und gleißender" Hinterlist „Die Mätresse des Bischofs" nannte, obwohl darin nicht einmal der Bischof eine Rolle spielt. Mit seinem ursprünglichen Arbeitstitel „Zwei Jahre Iberer-Forschung" hätte er wohl zu wenig Bücher abgesetzt. Und ganz ehrlich — hätten Sie bis hierhin durchgehalten, wenn Sie gewußt hätten, daß `Moose` lediglich ein geschwätziger Elch ist?

`Moose`-Autor Bill Perkins spürte den Elch irgendwann im Sommer 1987 auf einem Mac auf. Da durfte Amiga natürlich nicht hintenanstehen, und Bill machte sich spontan an die Arbeit. Das Ergebnis? Klicken Sie auf das `Moose`-Icon der RPD-Diskette, einige Sekunden später erscheint zum ersten Mal der Elch auf dem Bildschirm, sagt einen seiner zahlreichen dummen Sprüche auf und verschwindet wieder. Sollte `Moose` sich nicht aus der Reserve locken lassen, kann es daran liegen, daß sich sein Kehlkopf (das `narrator.device`) nicht im `devs`-Verzeichnis Ihrer Startdiskette befindet. Sie müßten

© 1990 technicSupport



Moose wie er leibt und lästert

es dann von einer vollständigen Workbench dorthin kopieren. Von nun an werden Sie keine ruhige Minute mehr haben, alle paar Sekunden meldet sich `Moose` mit einem neuen Spruch.

Glücklicherweise gibt es ein zweites Programm namens `MooseCtrl`. Es läßt sich nur aufrufen, wenn der Elch selbst aktiv ist, und erlaubt die Änderung einiger Parameter. Hier können Sie `Moose` Sprechverbot erteilen, was er allerdings mit einer weiteren unqualifizierten Äußerung quittieren wird. Alle Einstellungen werden in einem kleinen Eingabefenster mit der Maus getätigt:

`Male` `Moose` spricht mit männlicher Stimme.

`Female` Der Elch hat Kreide gefressen.

`Normal` `Moose` spricht normal — logisch!

`Robotic` Der Elch klingt, als ob er die Federn eines Ikea-Sofas verschluckt hat.

`On` `Moose` wird Redeerlaubnis erteilt.

`Off` `Moose` verkriecht sich im Speicher, bis er wieder etwas sagen darf.

`Pitch` Beeinflußt die Stimmlage.

`Rate` Regelt die Sprechgeschwindigkeit. Das Tempo reicht von „Willy Brandt — verschlafen" bis „Dieter-Thomas Heck — beim Ferngespräch nach Tasmanien".

`Samples` Bestimmt die Menge an verschluckter Kreide.

`Delay` Legt die Regenerationszeit für die Stimmbänder fest. `Moose` darf sich zwischen seinen Auftritten maximal 30 Minuten erholen.

Dieses Kontrollprogramm benötigt einen eigenen Zeichensatz, der sich im `Font`-Directory der RPD-Diskette befindet. Wenn Sie `Moose` auf eine andere Diskette installieren oder mit einer anderen Disk als der RPD booten möchten, dann müssen Sie diesen Zeichensatz in das entsprechende Verzeichnis der Bootdiskette kopieren, also z.B.:

```
copy df1:Fonts df0:Fonts all
```

wenn sich im Laufwerk `df1:` die RPD-Diskette befindet.

Erweiterung des Sprachschatzes

Abgesehen von einigen Kernsätzen, die im `Moose`-Programm selbst stehen, holt sich unser Elch seine lockeren Bemerkungen aus der Datei „Moose.Phrases", die unter exakt diesem Namen im Hauptverzeichnis der Diskette stehen muß, auf der sich auch `Moose` aufhält. Diese Datei muß genau 2^n (also 2, 4, 8, 16, 32, 64, etc.) Phrasen enthalten, wobei jede Phrase in einer eigenen Zeile stehen muß. Das Ende der Datei ist mit einer zusätzlichen Leerzeile oder einem *Form-Feed-Character* (ASCII=12) zu markieren. Wenn Sie eine solche Datei — z.B. mit dem Systemeditor `Ed` — erstellen, dürfen Sie die einzelnen Sätze nicht in ihrer normalen Schreibweise eintragen. Hier sind die *Phonemcodes* gefragt. Statt

```
Lets play a game
```

 © 1990 technicSupport

muß demnach

```
LEH4TS PLEY3 AH GEY6M
```

geschrieben werden. Diese Codes können Sie mit Hilfe des SAY-Tools von der „ExtrasD"-Diskette erhalten. Wenn Sie im unteren Fenster ein Wort oder einen Satz eingeben und [Return] drücken, wird der Satz ausgesprochen und im oberen Fenster erscheinen die Phonemcodes, die sich von dort abschreiben und dann in eine Datei übertragen lassen. Diese Methode ist allerdings recht mühsam und für deutsche Sprache nicht sehr gut geeignet.

„Speechd" als Dolmetscher

Auf derselben „ExtrasD"-Diskette findet sich im Verzeichnis Basic-Demos das Basic-Programm „Speechd", welches Texteingaben über die TRANSLATE$-Funktion so gut wie möglich in deutsch klingende Phonemfolgen umwandelt. Einige kleine Änderungen in diesem Programm genügen, und es erstellt aus unseren Vorgaben eine fertige Datei für Moose. Die folgende Lösung ist insofern als *„quick and dirty"* zu bezeichnen, als sie das Öffnen und Schließen der Ausgabedatei automatisch erfolgen läßt, weshalb es unbedingt nötig ist, das Programm nach der Arbeit durch die Eingabe von „Ende" zu verlassen, da die Ausgabedatei andernfalls nicht geschlossen wird. Für jede neue Datei muß das Programm neu aufgerufen werden.

Die Änderungen sind schnell gemacht! Öffnen Sie das Programm „Speechd" mit einem Doppelklick auf sein Icon. Nach dem Laden wählen Sie aus dem Run-Menü die Stop-Funktion und anschließend aus dem Windows-Menü die Funktion Show List. Es erscheint das Editierfenster, das auf volle Bildschirmbreite zu bringen ist. Klicken Sie mit dem Mauszeiger in dieses Fenster, und es erscheint ein Cursor. Mit den Cursortasten können Sie sich jetzt durch das gesamte Programmlisting bewegen. Ziemlich weit oben werden Sie die Zeile

```
COLOR 0,1:zeile=6
```

entdecken. Fügen Sie unterhalb dieser folgende neue Zeile ein:

```
OPEN "ram:Phrasen" FOR OUTPUT AS #1
```

Dieser Befehl öffnet beim Programmstart in der Ram-Disk eine Ausgabedatei namens „Phrasen“. Sie können aber auch jeden anderen Namen verwenden. Weiter unten finden Sie dann die Zeile

```
IF LEFT$(msgt$)="Ende" THEN SYSTEM
```

Hier wird das Programm verlassen. Vorher möchten wir noch mit einem *Form-Feed-Character* das Dateiende kennzeichnen und die Datei schließen. Dazu fügen wir zwei Befehle ein, so daß die Zeile folgende Gestalt erhält:

```
IF LEFT$(msgt$)="Ende" THEN PRINT #1,CHR$(12);:CLOSE #1:SYSTEM
```

Diese Befehlsfolge muß unbedingt in einer einzigen Zeile stehen. Betätigen Sie also nicht versehentlich die [Return] -Taste! Noch etwas tiefer steht schließlich die Zeile

```
SAY msgt$, current%
```

Diesen hier schon übersetzten String geben wir nun in unsere Datei aus. Dazu fügen wir unterhalb der Zeile eine weitere ein:

```
PRINT #1,msgt$
```

Jetzt wählen wir `Save As` aus dem `Project`-Menü, tragen dort einen neuen Namen (z.B. „SpeechMoose“) ein und speichern die geänderte Version ab. Diese können wir nun aufrufen, nacheinander 2, 4 oder eine andere erlaubte Satzanzahl eingeben und das Programm durch Eingabe von „Ende“ verlassen. Die fertige Datei kopieren wir aus der Ram-Disk auf unsere Elch-Disk und geben ihr gleich den richtigen Namen. Die alte Datei wird dadurch aber überschrieben:

```
copy ram:Phrasen df<x>:Moose.phrases
```

Jetzt können wir `Moose` aufrufen — er wird uns nach dem Munde reden.

PROBLEME

`Moose` läßt sich zwar ruhigstellen, nicht jedoch aus dem Speicher entfernen. Mit `MooseCtrl` getroffene Änderungen werden nicht gespeichert und sind nach dem Ausschalten des Rechners verloren.

 © 1990 technicSupport

Beim Aufruf lassen sich keine Parameter übergeben, was zumindest hinsichtlich der Erscheinungshäufigkeit wünschenswert wäre. Entgegen der Programmbeschreibung verlangt `Moose` einen Aufruf von der Workbench. Um das Programm auch aus dem CLI zu starten, kann man sich aber des Befehls „WBRun“ (z.B. auf Fishdisk 043) bedienen:

```
run WBRun > nil: Moose
```

Auf diese Weise kann man Freunden eine „nette“ Überraschung bereiten, wenn man bei ihnen das Programm heimlich installiert und den Aufruf in die Startup-Sequence hineinmogelt.

BEMERKUNGEN

`DOW4NT TEHL EH3NIYWAH1N AYM /HIY4R.`

HERKUNFT

Dieses Shareware-Programm findet man auf der RPD-Diskette 79. Der Autor wünscht sich eine Gabe von $5 bis $10. Für $15 (Europäer sollten etwas fürs Porto drauflegen) erhält man den kompletten, kommentierten Assembler-Code und — sofern schon fertig — einen Spracheditor. Die Anschrift:

```
Bill Perkins
142 Burban Drive
Branford, Ct. 06405
Compuserve: 72245, 710
```

MouseZoom

Ein etwas anderer Mausbeschleuniger

AUFRUF

Von der Workbench wird das Programm durch einen Doppelklick auf sein Icon aktiviert, ein zweiter Doppelklick stellt den Ausgangszustand wieder her. Im CLI genügt jeweils die Eingabe von

```
MouseZoom
```

um das Programm ein- bzw. auszuschalten. Beim Aufruf aus der `Startup-Sequence` ist die Verwendung von `Run` oder `RunBack` nicht erforderlich.

BESCHREIBUNG

Die ideale Mausgeschwindigkeit gibt es nicht. Bei einem Programm kommt es darauf an, den Mauszeiger besonders exakt zu positionieren, bei einem anderen müssen häufig längere Wege auf dem Bildschirm zurückgelegt werden. Das Anpassen der Mausgeschwindigkeit an die jeweiligen Anwendungen wird auf die Dauer zu einer lästigen Aktion. Bequemer hat man es da mit `MouseZoom`, einem nur 2520 Byte langen Programm, das seine große Funktionalität einem kleinen Trick verdankt.

Das Verhältnis zwischen der Strecke, um die die Maus bewegt wird, und der vom Mauszeiger auf dem Bildschirm zurückgelegten Strecke ist bei `MouseZoom` nicht konstant, sondern hängt wiederum von der Geschwindigkeit der Mausbewegung ab. Je schneller die Maus bewegt wird, desto größer wird auch die Beschleunigung, die der Mauszeiger durch das Programm erfährt. Mit langsamen Mausbewegungen kann man auf diese Weise sehr exakt positionieren, mit schnellen, aber dennoch kurzen Bewegungen große Strecken zurücklegen. An dieses neue Arbeitsgefühl hat man sich sehr rasch gewöhnt, das Ändern der Mausgeschwindigkeit ist dann kein Thema mehr.

HERKUNFT

`MouseZoom` stammt von `John Meissen` und ist mit seinem C-Quellcode (Lattice 4.00) auf der RPD 148 zu finden.

© 1990 technicSupport

Movie, Dilbm, Pilbm

Animationen mit weniger Speicherbedarf

AUFRUF

Die Programme `Dilbm` und `Pilbm` lassen sich nur aus dem CLI heraus aufrufen. Eine fertige Animationssequenz läßt sich vom CLI, aber auch von der Workbench starten, wenn sie mit einem Project-Icon versehen wird.

ZWECK

`Dilbm` und `Pilbm` erzeugen aus IFF-Bildern besonders kompakte Animationsdateien, die mit `Movie` abgespielt werden können.

BESCHREIBUNG

Die meisten Animationen arbeiten nach dem Prinzip des Daumenkinos: Mehrere IFF-Bilder, die sich jeweils nur geringfügig voneinander unterscheiden, werden in kurzen Zeitabständen nacheinander eingeblendet. Bei geschicktem Arrangement der Bilder gewinnt der Betrachter den Eindruck einer kontinuierlichen Bewegung. Da es wegen der kurzen Zeitintervalle in der Regel nicht möglich ist, Daten von der Festplatte oder den noch langsameren Laufwerken nachzuladen, müssen alle Bilder vor dem Starten einer Animation im RAM bereitstehen, woraus sich zwangsläufig ein großer Speicherbedarf ergibt.

Nach einem etwas abgewandelteten Verfahren arbeiten die hier beschriebenen Programme: Eine mit `Dilbm` und `Pilbm` aufbereitete Animationssequenz besteht nicht mehr aus kompletten Bildern, sondern aus lediglich einem vollständigen Anfangsbild. Alle weiteren Bilder werden aus diesem Ausgangsbild berechnet, wobei nur die Unterschiede zwischen jeweils zwei Bildern berücksichtigt werden. Dieses Verfahren ist besonders rationell und benötigt vergleichsweise wenig Speicherplatz. Die maximale Abspielgeschwindigkeit beträgt 60 Bilder pro Sekunde. Auch HAM-, Overscan- und PAL-Bilder werden verarbeitet. Die Bilder müssen jedoch von gleicher Größe sein und dieselbe Farbpalette besitzen.

© 1990 technicSupport

Bevor man sich die Mühe macht, eine eigene Animation zu planen und mit einem Malprogramm oder per Ray Tracing eine Vielzahl von Bildern zu erstellen, sollte man die drei Programme erst einmal ausprobieren. Dazu greifen wir auf eine bereits fertige Bildsequenz zurück, die sich in der `QuickFlix/Swing`-Schublade der Fishdisk 106 befindet. Es ist eine sehr einfach gehaltene Bilderfolge, die einige schwingende Pendel darstellt. Am besten legt man sich eine spezielle Diskette an, auf der man die drei Programme und die benötigten Bilder unterbringt:

```
format drive df1: name Anim noicons
cd Ruhrsoft-Public_Domaine_79:
copy (dilbm|pilbm|movie) Anim:
cd AmigaLibDisk106:Quickflix/Swing
copy ball#? Anim:
```

Da die Bezeichnung der Bilder etwas verwirrend ist, sollten sie umbenannt werden:

```
cd Anim:
rename ball000 bild1
rename ball100 bild2
rename ball200 bild3
rename ball300 bild4
rename ball001 bild5
rename ball002 bild6
rename ball003 bild7
```

Auf der Diskette befinden sich nun sieben Bilder, die in einer bestimmten Reihenfolge abgespielt werden müssen, um den Schwingungsvorgang korrekt wiederzugeben. Diese Reihenfolge lautet: 1, 2, 3, 4, 3, 2, 1, 5, 6, 7, 6, 5, 1, 2. Daß die Bildsequenz auf denselben zwei Bildern endet, mit denen sie begonnen hat, ist eine wichtige Voraussetzung dafür, daß die Sequenz später in Form einer Schleife abgespielt werden kann.

Die Differenzen berechnen: Dilbm

Der nächste Arbeitsschritt besteht darin, mit dem Programm `Dilbm` sogenannte Delta-Dateien zu erzeugen, die die Differenz zwischen

 © 1990 technicSupport

jeweils zwei Bildern beschreiben. Dazu muß man das Prinzip des Abspielens etwas genauer kennen. Das `Movie`-Programm lädt das erste Bild komplett in den Speicher und berechnet dann das zweite Bild aus einer Delta-Datei, die den Unterschied zwischen Bild 1 und Bild 2 beschreibt. Von diesem Zeitpunkt an stehen jederzeit zwei Bilder im Speicher, von denen eines gezeigt wird, während aus dem anderen das nächste Bild berechnet wird. Während Bild 2 zu sehen ist, wird also aus Bild 1 das Bild 3 erzeugt. Anschließend erscheint Bild 3 und Bild 2 wird in Bild 4 umgewandelt usw. Ebenfalls von Bedeutung ist, daß eine Delta-Datei normalerweise nur in einer Richtung arbeitet, Bild 1 also in Bild 3 umwandeln kann, aber nicht umgekehrt. Soll die Umwandlung in beiden Richtungen möglich sein, kann dem `Dilbm`-Befehl jedoch die Option `X` mitgegeben werden, was man bei vielen Anwendungen ausnutzen kann, um die Zahl der Delta-Dateien niedrig zu halten.

Jetzt läßt sich eine Tabelle erstellen, aus der man ablesen kann, welche Delta-Dateien für die Animation benötigt werden:

```
bild1   bild1     Ausgangsbild
bild2   delta1-2  aus Bild1 und Bild2
bild3   delta1-3  aus Bild1 und Bild3
bild4   delta2-4  aus Bild2 und Bild4
bild3   delta3-3  aus Bild3 und Bild3
bild2   delta4-2  gleich delta2-4 (x-Option)
bild1   delta3-1  gleich delta1-3 (x-Option)
bild5   delta2-5  aus Bild2 und Bild5
bild6   delta1-6  aus Bild1 und Bild6
bild7   delta5-7  aus Bild5 und Bild7
bild6   delta6-6  aus Bild6 und Bild6
bild5   delta7-5  gleich delta5-7 (x-Option)
bild1   delta6-1  gleich delta1-6 (x-Option)
bild2   delta5-2  gleich delta2-5 (x-Option)
```

Der Aufruf des `Dilbm`-Kommandos ist einfach, er erfordert drei Dateinamen als Argumente: den Namen des jeweiligen Ausgangsbildes, den des zu berechnenden Bildes sowie den Namen für die Delta-Datei. Der Befehl kommt jedoch mit der voreingestellten Stack-

Größe nicht aus, diese sollte auf mindestens 17000 erhöht werden, da sich andernfalls der Guru zu Wort meldet:

```
stack 17000
```

Mit dem folgenden Befehl läßt sich nun die erste Delta-Datei erzeugen:

```
dilbm bild1 bild2 delta1-2
```

Auf dem Bildschirm erscheint dabei ein Bild, welches die Unterschiede zwischen den Einzelbildern verdeutlicht. Die Anzeige dieses Bildes läßt sich mit der `Q`-Option unterdrücken, was die Ausführung des Befehls beschleunigt. Da man leicht die Übersicht verliert, wenn man alle Befehle nacheinander eingibt, und immer auf den Rechner warten muß, ist es praktischer, mit dem `Ed` eine Batchdatei zu erzeugen, etwa mit dem Befehl

```
ed dilbm.script
```

und sie dann mit

```
execute dilbm.script
```

aufzurufen. Diese Datei muß aus folgenden Zeilen bestehen:

```
dilbm bild1 bild2 delta1-2   q
dilbm bild1 bild3 delta1-3 x q
dilbm bild2 bild4 delta2-4 x q
dilbm bild3 bild3 delta3-3   q
dilbm bild2 bild5 delta2-5 x q
dilbm bild1 bild6 delta1-6 x q
dilbm bild6 bild6 delta6-6   q
dilbm bild5 bild7 delta5-7 x q
```

Die Berechnung der Delta-Dateien ist recht zeitaufwendig, daher kann eine kurze Kaffeepause eingelegt werden.

 © 1990 technicSupport

Verbinden der Einzeldateien: Pilbm

Das Programm `Pilbm` hat nun die Aufgabe, aus den Rohmaterialien eine einzige Datei zu bilden. Alle Informationen, die hierzu nötig sind, müssen in einer Datei abgelegt werden:

```
ed Pendel.script
```

Diese Datei erfordert ein ganz bestimmtes Format: In Zeile 1 gehört der Name der Animationsdatei, die erzeugt werden soll, Zeile 2 enthält den Namen des Ausgangsbildes (in unserem Fall `bild1`) und Zeile 3 den Namen der Delta-Datei, die aus dem Ausgangsbild das zweite Bild formt. In den nun folgenden Zeilen erscheinen die Namen der übrigen Delta-Dateien in beliebiger Reihenfolge. Dieser Deklarationsteil wird durch ein Sternchen abgeschlossen. Hier das Beispiel für unsere Animation:

```
Pendel      ; Name der Amimation
bild1       ; Ausgangsbild
delta1-2    ; Delta-Datei für 2. Bild
delta1-3    ; Delta-Datei 1
delta2-4    ; Delta-Datei 2
delta3-3    ; Delta-Datei 3
delta2-5    ; Delta-Datei 4
delta1-6    ; Delta-Datei 5
delta6-6    ; Delta-Datei 6
delta5-7    ; Delta-Datei 7
*
```

Die Zeichen ab dem Semikolon sollten nicht abgetippt werden, sie dienen nur zur Verdeutlichung des Aufbaus. An diese Zeilen schließt sich der Anweisungsteil, der die Reihenfolge der Bilder und die Zeitabstände festlegt. Jede Zeile enthält zu diesem Zweck ein Paar aus zwei Ziffern, deren erste eine Delta-Datei durch ihre Position im Deklarationsteil bestimmt: Auf die Datei `delta2-5` wird beispielsweise durch die Ziffer 4 Bezug genommen. Die zweite Ziffer definiert den zeitlichen Abstand zwischen zwei Bildern, je kleiner der Wert, desto schneller wird die Animation abgespielt. Diese Werte entsprechen keinen physikalischen Zeiteinheiten, ihre Wirkung läßt sich am besten durch Ausprobieren ermitteln. Um die Delta-Dateien in der

Reihenfolge, wie sie aus der obersten Tabelle ermittelt wurde, aufzurufen, muß der Anweisungsteil die folgende Gestalt annehmen:

```
1 1
2 1
3 1
2 1
1 1
4 1
5 1
7 1
6 1
7 1
5 1
4 1
1 -1
```

Der Zeitfaktor wurde konstant auf 1 gesetzt, um einen gleichmäßigen Ablauf zu gewährleisten. Eine Besonderheit zeigt sich in der letzten Zeile. Nimmt der Zeitfaktor einen negativen Wert an, so wird die entsprechende Zeile als Sprunganweisung zu der durch die erste Ziffer bestimmten Zeile des Anweisungsteils gewertet. In diesem konkreten Fall wird zu Zeile 1 zurückgesprungen, was bedeutet, daß der gesamte Anwendungsteil in einer Endlosschleife abgearbeitet wird.

Nach der Fertigstellung der Scriptdatei, kann sie mit dem Befehl

```
pilbm Pendel.script
```

bearbeitet werden, was sehr schnell erfolgt. Auf der Diskette findet man nun die Datei `Pendel`, die nur 16806 Bytes lang ist. Mit der Eingabe

```
Movie Pendel
```

läßt sich die Animation abspielen. `Movie` öffnet zunächst ein Fenster mit einer Programminformation und einem Copyright-Hinweis. Nach Drücken einer beliebigen Taste wird die Bildsequenz geladen und abgespielt. Dieser Vorgang kann jederzeit mit [Esc] abgebrochen werden.

 © 1990 technicSupport

Die fertige Animation läßt sich workbenchgerecht aufbereiten, wenn die Datei `Pendel` mit einem Project-Icon versehen wird. Dieses Icon ist dann auf der Workbench einmal anzuklicken, anschließend ist die Funktion Info anzuwählen. In das Feld für die Stack-Größe ist ein Wert von 17000 einzutragen, in das Feld für das Default Tool gehört der Name des Movie-Programms einschließlich der Pfadangabe. Für die hier beschriebene Diskette lautet der Eintrag „`:movie`", sofern sich die Datei `Pendel` noch auf derselben Diskette befindet. Nach dem Verlassen des Info-Fensters über den [Save] -Button kann die Animation auch per Doppelklick auf das Pendel-Icon gestartet werden.

PROBLEME

Der Versuch, `Dilbm` auf eine weitere Bildsequenz anzuwenden, die sich im `lmv`-Verzeichnis von Fishdisk 73 befindet, ergab Delta-Dateien, die größer waren als die eigentlichen Bilder. Der Grund für dieses Verhalten ist darin zu sehen, daß es sich hier um digitalisierte Bilder handelt, die sich zwar für das Auge kaum voneinander unterscheiden, während der Computer bei fast jedem Bildpunkt einen ganz leichten Unterschied in den Grauwerten zweier Bilder feststellt. In solchen Fällen ist der Einsatz der drei beschriebenen Programme nicht effektiv und daher auch nicht sinnvoll. Probleme gibt es ebenfalls, wenn die Bilder unterschiedliche Größen und Farbpaletten aufweisen. Das Manual warnt auch noch davor, bei der Benutzung von `Movie` andere Utilities im Speicher zu halten. Diesbezüglich konnten wir jedoch keine Schwierigkeiten feststellen.

HERKUNFT

Die hier besprochene Version 1.01 ist auf der RPD-Disk 079 zu finden. Zu verdanken ist sie den Schöpfern von `Sculpt 3D` und `Sculpt-Animate: Eric Graham` und `Byte by Byte corporation, (512) 343-4357`.

VERWEISE

`DPSlide` (Fishdisk 011), `show` (Fishdisk 060), `SlideShow` (Faugdisk 038), `Lmv` (Fishdisk 073), `QuickFlix` (Fishdisk 106), `SuperView` (TBAG 25), `ShoWiz` (TBAG 31)

© 1990 technicSupport

NECquik

Druckerkontrolle von der Workbench

AUFRUF

`NECquick` läßt sich durch Anklicken seines Icons oder aus dem CLI heraus durch Eingabe des Programmnamens starten.

ZWECK

`NECquik` ermöglicht die Kontrolle der Funktionen eines NEC-kompatiblen Druckers von der Workbench aus.

BESCHREIBUNG

Statt einer ausführlichen Beschreibung findet der Leser an dieser Stelle nur eine kurze Vorstellung der Fähigkeiten dieses Programms, das bereits mit einer deutschen Anleitung versehen ist.

`NECquik` ist mit dem ebenfalls in diesem Buch vorgestellten `Print-Pop` zu vergleichen, jedoch erweist es sich als wesentlich leistungsfähiger, da es neben Druckqualität, Zeichenweite und Rändern noch weitere Parameter beeinflussen kann: Fettdruck, Kursivdruck, Unterstreichen, Schattendruck (falls vorhanden), benutzerdefinierten Zeichensatz, vertikale und horizontale Vergrößerung, Zeilenvorschub, Zeilenabstand und Druckart. In ein weiteres Gadget läßt sich ein Text eingeben, der zum Drucker geschickt wird, ein anderes ermöglicht die hexadezimale Eingabe einer druckereigenen Steuersequenz, so daß sich auf diesem Weg jede Druckerfunktion ansprechen läßt. Der Übertragungskanal zum Drucker läßt sich vom Programm aus unterbrechen, so daß der Drucker jederzeit von anderen Programmen benutzt werden kann, ohne daß `NECquik` beendet werden muß. Diese Fähigkeiten machen `NECquik` zu einem hervorragenden Hilfsmittel für alle Besitzer eines NEC-kompatiblen Druckers.

Jedoch konnte diese Funktionsvielfalt nur unter Umgehung des `printer.device` realisiert werden: `NECquik` sendet seine Daten daher über die parallele Schnittstelle (`par:`), nicht aber über das `prt:`-Gerät. Wer einen an der parallelen Schnittstelle angeschlossenen,

© 1990 technicSupport

```
NECquik © '88 Tim Pietzcker
Schrift:  Schön        Druckweite :   10 cpi
Fettdruck : aus        Schattendruck  :  aus
Kursiv    : aus        Horiz.Vergrößerung: 0
Unterstrich:aus        Vert. Vergrößerung: 0
User Font : aus        Spezial :         aus
Druckart      0        Zeilenabstand 25 /180"
Ränder  1  80          Vorschub   5   Zeilen
 Drucker-Reset         Hexacode 1B 70 01
Schicke Text zum Drucker !
Clr          Drucker  :  an              OK
```

Mit NECquik bekommt man NEC-Drucker unter Kontrolle

aber nicht NEC-kompatiblen Drucker besitzt, wird die Funktionen dieses Programmes daher nur bedingt nutzen können. In diesem Fall muß er selbst ermitteln, welche Funktionen auch auf dem eigenen Drucker verfügbar sind. Programmierern mit Grundkenntnissen in der Sprache C dürfte es jedoch nicht schwer fallen, den mitgelieferten Programmcode entsprechend abzuändern. Wenn man keine derartigen Kenntnisse oder keinen Compiler besitzt, dann kann man immer noch den Diskettenmonitor `Newzap` — ebenfalls in diesem Band besprochen — zu Hilfe nehmen, um das ausführbare `NECquik`-Programm zu patchen. Dabei sind die im Programm enthaltenen Steuersequenzen durch die für den eigenen Drucker gültigen zu ersetzen. Dazu muß man sich nicht einmal die Werte für den NEC-Drucker besorgen; diese lassen sich nämlich auch aus dem gut dokumentierten Programmcode `NECquik.c` herauslesen. Durch Patchen (indem man im Programmcode `par:` durch `ser:` ersetzt) läßt sich auch erreichen, daß das Programm mit Druckern an der seriellen Schnittstelle zusammenarbeitet. Da uns kein serieller Drucker zur Verfügung stand, haben wir die Ausgaben einer auf diese Weise

gepatchten Programmversion zur Überprüfung mit dem `Cmd`-Kommando der Workbench 1.3 in eine Datei umgeleitet und konnten feststellen, daß die Übertragung ordnungsgemäß funktionierte.

HERKUNFT

Dieses Programm befindet sich auf der `Franz 45`. Die Adresse des Autors:

`Tim Pietzcker`
`Schwarzwaldstr.52`
`D-7819 Denzlingen`

VERWEISE

`PrintPop` (Panorama 23c), `gwPRINT` (RPD 160)

© 1990 technicSupport

NewZap

Dateiorientierter Diskettenmonitor

AUFRUF

`NewZap` läßt sich von der Workbench aus durch Anklicken des Icons starten. Aus dem CLI heraus lautet der Aufruf:

`NewZap3.18` [⟨*Dateiname*⟩]

Zur bequemeren Handhabung empfiehlt es sich, das Programm in den Namen `NewZap` umzubenennen.

OPTIONEN

⟨*Dateiname*⟩ Der Name der Datei, die mit dem Diskettenmonitor bearbeitet werden soll. Befindet sich die Datei nicht im aktuellen Verzeichnis, so muß zusätzlich der komplette Pfadname angegeben werden.

ZWECK

`NewZap` ist ein Programm, mit dem sich auf einfache Weise Dateien, darunter auch ablauffähige Programme, manipulieren und an eigene Bedürfnisse anpassen lassen.

BESCHREIBUNG

Die Aufgabe eines Diskettenmonitors läßt sich mit wenigen Worten beschreiben: Er liest Daten von einer Diskette oder einem vergleichbaren Speichermedium, stellt sie auf dem Bildschirm zur Bearbeitung bereit und schreibt die geänderten Daten, wenn es gewünscht wird, wieder auf die Diskette zurück. Viele dieser Diskettenmonitore orientieren sich dabei an der physikalischen Struktur einer Diskette, d. h. an den einzelnen Spuren und Sektoren. Der Vorteil dieser Arbeitsweise besteht darin, daß der Zugriff auf jeden beliebigen Block (z. B. Rootblock, Bootblock) möglich ist. Den Manipulationsmöglichkeiten sind also keine Grenzen gesetzt. Auf der anderen

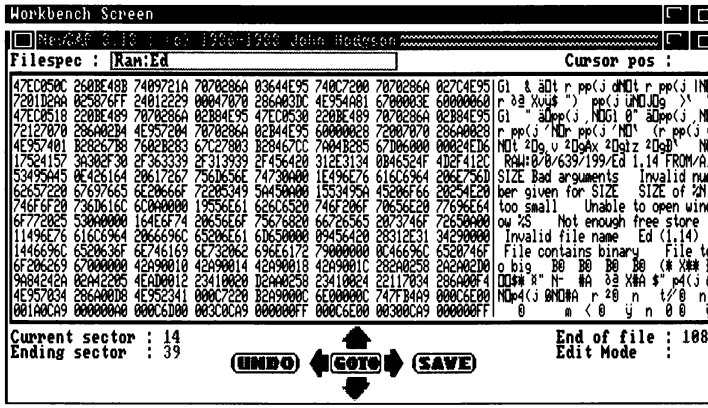
Seite erfordert der Umgang mit einem dieser Monitore eine recht fundierte Kenntnis über die Diskettenstruktur des AmigaDOS.

In vielen Fällen geht es auch einfacher. Dann nämlich, wenn die Daten, die geändert werden sollen, in Form einer Datei vorliegen. Wenn es sich um reine Textdateien handelt, genügt dazu natürlich ein normaler Texteditor. Oftmals ist man jedoch versucht, Programmdateien zu ändern, deren Hieroglyphen ein Texteditor gewöhnlich nicht in adäquater Form darstellen kann. Da die Maschinencodes solcher Programme nur unter allergrößtem Aufwand entschlüsselt werden können, beschränken sich die Manipulationen in Programmen meistens auf das Verändern bestimmter Informationen, die in Klartextform in diesen Programmen enthalten sind.

In diesen Fällen leistet ein Diskettenmonitor wie `NewZap` gute Dienste. Er orientiert sich nicht an der physikalischen Struktur der Disketten, sondern an Dateien, die als in sich geschlossene Einheiten bearbeitet werden können, ohne daß man sich über die Diskettenstruktur Gedanken machen muß. Blöcke, die andere Datenstrukturen enthalten, können von `NewZap` allerdings nicht verarbeitet werden.

Nach dem Aufruf öffnet `NewZap` ein eigenes Fenster, das in der Hauptsache aus zwei großen Feldern besteht, in denen die einzelnen Bytes einer Datei angezeigt werden. Unterhalb der Menüzeile liegt ein Gadget mit der Bezeichnung `Filespec:` Nach dem Anklicken kann in dieses Feld der Name (mit Pfadangabe) der gewünschten Datei eingetragen werden. Diese Datei wird dann geladen und angezeigt. Das linke, größere Feld zeigt die Bytewerte in hexadezimaler Darstellung, das rechte Feld dieselben Werte in der ASCII-Textdarstellung. Ober- und unterhalb dieser Anzeigefläche werden noch einige Status-Informationen gegeben. `CurPos` bezeichnet die Position des Textzeigers, `Current sector` die Nummer des angezeigten Blocks, `Ending sector` die Gesamtzahl der Blöcke, die die Datei belegt. Diese Blocknummern dienen lediglich der besseren Orientierung des Anwenders, sie haben nicht mit den Blöcken zu tun, die die Datei auf der Diskette belegt. `Edit Mode` zeigt, nachdem man in eines der Anzeigefelder geklickt hat, den aktiven Editiermodus an: Steht die Anzeige auf `HEX` so wirken die Änderungen auf die

 © 1990 technicSupport



NewZap verleiht Ed die volle Bildschirmgröße

Hex-Darstellung, steht sie auf ASC, so wirken die Änderungen auf das rechte Feld.

Aber betrachten wir die grundsätzliche Bedienung an einem typischen Beispiel! Wenn man ab und zu mit dem Ed arbeitet, wird man sich schon darüber geärgert haben, daß dieser Editor sein Fenster nicht über den vollen Bildschirm, sondern nur bis Zeile 199 öffnet. Um dies zu ändern, trägt man in das Filespec-Gadget

```
C:Ed
```

ein und drückt die [Return] -Taste, waraufhin der erste Block der Datei Ed geladen und angezeigt wird. Durch Anklicken der vier Pfeil-Symbole kann man sich nun blockweise durch die Datei bewegen. F steht dabei für vorwärts, B für rückwärts, S springt in den ersten, E in den letzten Block einer Datei. Klickt man in das [GOTO] -Feld, erscheint ein Textzeiger, und man kann durch Eingabe einer Nummer einen gewünschten Block direkt anspringen. Beim Durchblättern einer Datei entdeckt man im rechten Fenster von Zeit zu Zeit les-

bare Texte, zumeist handelt es sich dabei um Fehler- und Statusmeldungen des Programmes. Für unser Vorhaben ist Sektor 14 interessant. Hier entdeckt man an der Cursor-Position `0A1` den Text `RAW:0/0/639/199`. Dieser Ausdruck definiert die Größe des Fensters, das beim Programmaufruf geöffnet wird. Nun braucht man lediglich mit dem Mauszeiger auf die „1" von „199" zu klicken (Cursorposition = `0AD`, der Eingabemodus wird dann automatisch auf `ASC` gestellt) und diesen Wert mit der Ziffernfolge „255" zu überschreiben. Falsche Eingaben können durch einen Klick auf den [UNDO] -Button rückgängig gemacht werden. Ist der gewünschte Wert eingetragen, wird die geänderte Datei mit [SAVE] wieder auf die Diskette zurückgeschrieben. Da man Dateinamen und Pfad nicht ändern kann, wird die alte Datei gnadenlos überschrieben, weshalb man solche Änderungen zunächst nur an Sicherheitskopien erproben sollte. Ruft man nun den `Ed` auf, dann nutzt er den vollen PAL-Bildschirm aus. Ein weiteres Beispiel für die Verwendung von `NewZap` läßt sich im Kapitel über die Workbench 1.3 für Invidualisten nachlesen.

Die Menüleiste

Die Menüleiste gliedert sich in drei Gruppen. Das `Project`-Menü ermöglicht das Öffnen eines CLI's und das Verlassen des Programms. Das `Move`-Menü stellt dieselben Funktionen bereit, die sich auch über die Pfeil-Symbole aufrufen lassen. Interessanter ist da das `Search`-Menü, das folgende Kommandos enthält:

`String Search`	Es öffnet sich ein Requester, in den ein Suchtext eingegeben werden kann, nach dem die Datei anschließend durchsucht wird. Bei Verwendung hexadezimaler Werte ist dem Suchbegriff ein Dollarzeichen ($) voranzustellen. Beim ersten Auftreten des Begriffs wird die Suche abgebrochen.
`Case Dependency`	Ermöglicht auf Wunsch die Unterscheidung zwischen Groß- und Kleinschreibung.
`Set Direction`	Die Suchrichtung kann von vorwärts auf rückwärts geschaltet werden.

 © 1990 technicSupport

`Continue Search` Nach dem ersten Auffinden des Suchbegriffs kann die Suche mit dieser Option fortgesetzt werden.

Vorsicht ist die Mutter...

Die einfache Bedienung des `NewZap`-Editors täuscht über die Gefahren hinweg, die mit dem Ändern ausführbarer Programme verbunden sind. Das Ändern von Texten ist die harmloseste Spielart, aber auch hier können schon folgenschwere Fehler entstehen. Ersetzt man nämlich einen bestehenden Text gegen einen längeren, können andere Bestandteile des Programms überschrieben werden, was im Extremfall das gesamte Programm ruinieren kann. Das bedeutet, daß man beispielsweise beim Eindeutschen von Menübezeichnungen mit dem vorgegebenen Platz auskommen muß.

Im linken Anzeigefeld lassen sich (im HEX-Modus) auch andere Programmteile verändern. Solche Manipulationen sollten nur von sehr erfahrenen Anwendern vorgenommen werden, da hierbei die Wahrscheinlichkeit eines schweren Fehlers sehr nahe bei 100% liegt.

HERKUNFT

`NewZap` ist ein Shareware-Programm, das man in der hier besprochenen Version 3.18 auf der Fishdisk 164 findet. Für $15 wird man eingetragener Benutzer. Die Anschrift des Autors:

```
Proprietary Products
c/o John Hodgson
P.O. Box 1643
Pebble Beach, Ca. 93953-1643
```

VERWEISE

`DiskX` (Fishdisk 158), `SmartDisk` (Panorama 28d), `DisKey` (Franz 46)

Paranoid

Erstklassiges Breakout-Spiel mit Level-Editor

AUFRUF

Wenn man mit der `Paranoid`-Diskette bootet, gelangt man über ein Titelbild mit Musikuntermalung direkt in das Menü.

ZWECK

`Paranoid` ist ein Geschicklichkeitsspiel, welches sich durch gute Grafik und guten Sound auszeichnet.

BESCHREIBUNG

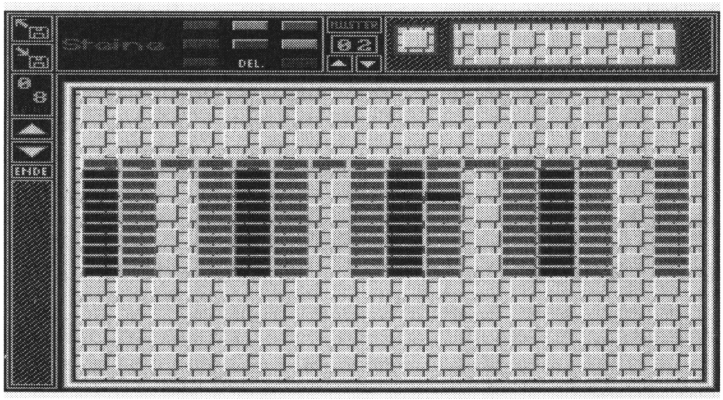
Breakoutspiele gibt es mittlerweile wie Sand am Meer. Auch in der Public Domain gibt es Varianten dieses Spieles (`SuperBrickout`, `3DBreakOut`), die jedoch von `Paranoid` bei weitem übertroffen werden. Der Abstand zwischen `Paranoid` und den kommerziellen Breakout-Versionen ist nicht mehr besonders groß. Aber nun zum Spiel selbst. Aus dem Menü heraus kann man mit der Maus wählen, ob man spielen, eigene Spielstufen entwerfen oder sich die Bestenliste ansehen möchte. Normalerweise wird man sich wohl für das Spiel entscheiden.

Das Spielprinzip

`Paranoid` öffnet einen Bildschirm, dessen größter Teil von der Spielfläche ausgefüllt wird. Auf dieser Spielfläche befinden sich farbige Steine, deren Anordnung von Spielstufe zu Spielstufe wechselt. Am unteren Rand des Feldes befindet sich ein Schläger, der mit der Maus von links nach rechts bewegt werden kann. Ziel des Spieles ist es, einen Ball im Spielfeld zu halten und alle Steine abzuräumen, um dann in die nächste Spielstufe zu gelangen.

Mit einem Druck auf die linke Maustaste wird der Ball abgeschlagen. In der rechten unteren Ecke befindet sich ein Quadrat mit einem Fragezeichen, das sich von Zeit zu Zeit zu drehen beginnt.

© 1990 technicSupport



Der Level-Editor sorgt für längeren Spielspaß

Bewegt man dann den Schläger auf dieses Quadrat, so wird eine Sonderfunktion ausgelöst, z. B. wird der Schläger breiter oder der Ball langsamer. Es kann jedoch auch passieren, daß der Ball unberechenbare Flugbahnen einnimmt.

Oberhalb des Spielfeldes werden die Punktzahl, die Spielstufe und die Anzahl der „Leben" angezeigt. Die Zahl der Leben reduziert sich jedesmal, wenn man den Ball verpaßt.

Der Level-Editor

Eine sehr schöne Sache ist der Editor, mit dem jede Spielstufe nach eigenen Vorstellungen gestaltet werden kann, wozu nur die Maus benötigt wird. Durch Anklicken der [△] - und [▽] -Felder kann die Nummer der zu bearbeitenden Spielstufe herauf- bzw. herabgezählt werden. Das angewählte Level kann über die Diskettensymbole in der linken oberen Ecke geladen bzw. abgespeichert werden. Über die beiden anderen [△] - und [▽] -Felder läßt sich eines von

16 Hintergrundmustern auswählen. Das angewählte Muster wird durch Anklicken des [Muster] -Feldes geladen, dabei werden jedoch alle Steine gelöscht. Steine werden positioniert, indem das entsprechende Symbol angeklickt und anschließend innerhalb des Spielfeldes an die gewünschten Stellen geklickt wird. Verlassen wird der Editor über das [Ende] -Gadget.

Die Bestenliste

Bei genügend hoher Punktzahl darf man sich in der Bestenliste verewigen. Dazu klickt man mit dem Mauszeiger auf die gewünschten Buchstaben, die dann in die Tabelle übernommen werden. Mit [←] kann ein Buchstabe gelöscht werden. Mit [↵] wird die Highscore-Liste verlassen.

HERKUNFT

Das Programm `Paranoid` wurde von

`Felix Wente, Pappelweg 68, 6430 Bad Hersfeld`

geschrieben. Es befindet sich mit seinem Quellcode in C (kurze Assembler-Einschübe) auf Diskette Nr. 13 der Schatztruhe von Stefan Ossowski.

© 1990 technicSupport

PLAY

Jukebox für SONIX-Musikstücke

AUFRUF

Um den Abspielvorgang von der Workbench zu starten, sind einige Vorbereitungen nötig, die weiter unten besprochen werden. Vom CLI aus gibt es zwei verschiedene Möglichkeiten, `PLAY` aufzurufen:

`Play` [*] [I=⟨*Instrumentenverzeichnis*⟩] ⟨*Titel*⟩[.smus]

oder

`Play` [*] P=⟨*Befehlsdatei*⟩

OPTIONEN

* `PLAY` wird als Hintergrundtask gestartet. Diese Option empfiehlt sich, wenn das Fenster, aus dem heraus der Aufruf erfolgte, anschließend geschlossen oder ein Abbruch mit [CTRL]-[C] verhindert werden soll.

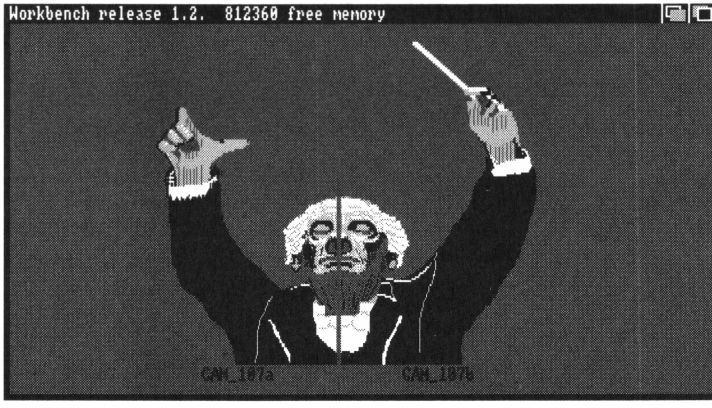
`Instrumentenverzeichnis`
Hier geben Sie den Namen des Verzeichnisses an, aus dem `PLAY` die erforderlichen Instrumente laden soll. Wenn dieses Verzeichnis nicht im aktuellen Verzeichnis liegt, müssen Sie eine Pfadangabe voranstellen.

`Titel` An dieser Stelle ist der Titel des Musikstücks gefragt, gegebenenfalls wieder mit Angabe des Pfades. Bei Angabe des Titels darf die Endung „.smus“ entfallen.

`Befehlsdatei`
Der Name einer Datei, die weitere Instruktionen enthält. Auf die Struktur solcher Dateien werden wir noch zu sprechen kommen.

ZWECK

`PLAY` ermöglicht es, mit dem bekannten Programm `SONIX` erstellte Musikstücke auch ohne dieses Programm und sogar noch komfor-



RPD 107A und 107B: Icons veraten den Inhalt

tabler abzuspielen. Mit Hilfe von PLAY können SONIX-Besitzer beispielsweise Demo-Disketten ihrer eigenen Kompositionen erstellen, die sie auch weitergeben dürfen. Über eine Befehlsdatei lassen sich auch längere Abfolgen verschiedener Stücke programmieren, die sich hervorragend zur Untermalung von Slide-Shows eignen. Jedoch dürfen Sie solche Slide-Shows oder Demos nicht für kommerzielle Zwekke nutzen, ohne vorher eine Lizenz vom Autor erhalten zu haben.

BESCHREIBUNG

Laut Programmbeschreibung sollten zu PLAY insgesamt 6 Elemente gehören:

`Play`	Das Hauptprogramm
`Play.info`	Ein Icon für das Hauptprogramm
`Play.doc`	Die Programmbeschreibung
`Scores.info`	Ein Muster-Icon für die Titel-Schublade

 © 1990 technicSupport

`Score.info` Ein Muster-Icon für Musiktitel

`Script.info` Ein Muster-Icon für Befehlsdateien

Auf RPD 107A fehlt leider das Icon für die Befehlsdateien. Das Icon für die Musiktitel findet sich nicht unter der angegebenen Bezeichnung, es trägt schon die Namen der vorhandenen Musikstücke.

Starten von PLAY über das CLI

Um das Programm bequem vom CLI aus aufrufen zu können, sollten Sie `PLAY` in das c-Verzeichnis Ihrer Arbeitsdiskette kopieren. Wenn Sie die Hinweise unter Aufruf und Optionen beachten, sollte der Aufruf keine Schwierigkeiten machen. Hier ein Beispiel, um das Musikstück „InTheMixPart1" von RPD-Diskette 107A zu starten.

```
cd CAM_107A:
Play I=Mix_Instruments Scores/InTheMixPart1
```

Wenn Sie `PLAY` nicht als Hintergrundprozeß aufrufen, läßt sich das Abspielen jederzeit über die Tastenkombination [CTRL]-[C] abbrechen. Wesentlich leistungsfähiger, aber auch entsprechend komplizierter ist das Abspielen mehrerer Stücke hintereinander über eine Befehlsdatei. Eine solche Befehlsdatei muß zunächst mit einem Editor erstellt und unter einem frei wählbaren Namen abgespeichert werden. Innerhalb der Befehlsdatei müssen einzelne Befehle durch [Space], [,] oder [Return] voneinander getrennt werden. Folgende Befehle stehen Ihnen zur Verfügung:

`D=`⟨*Verzeichnisname*⟩
: Setzt das aktuelle Verzeichnis

`I=`⟨*Instrumentenverzeichnis*⟩
: Definiert das Verzeichnis, in dem die Instrumente gesucht werden.

⟨*Titel*⟩ Ein Stück wird geladen und gespielt. Die Abarbeitung der Befehlsdatei wird bis zum Ende dieses Titels unterbrochen.

`A=`⟨*Titel*⟩ Ein Musikstück wird geladen. Noch während es gespielt wird, wird das nächste Stück nachgeladen.

V=⟨*Lautstärke*⟩ Durch Eingabe einer Zahl zwischen 0 und 255 legen Sie die Lautstärke fest. Bei Angabe von -1 wird die Originallautstärke der Stücke übernommen.

C=⟨*Häufigkeit*⟩ Diese Zahl gibt an, wie oft ein einzelner Titel wiederholt werden soll. Ein Wert von 0 steht für eine Wiederholung ohne Ende.

R=⟨*Blockwiederholung*⟩
Sie können dafür sorgen, daß eine bestimmte Kommandofolge innerhalb der Befehlsdatei beliebig oft wiederholt wird, indem Sie diese Option voranstellen. Das Ende der Kommandofolge muß durch einen Punkt gekennzeichnet sein. Ein Wert von Null steht auch hier für eine Endlosschleife.

;⟨*Kommentar*⟩ Wenn Sie ein Semikolon setzen, können Sie den Rest der Eingabezeile für Anmerkungen nutzen.

P=⟨*Befehlsdatei*⟩ Veranlaßt die Abarbeitung einer weiteren Befehlsdatei.

Das folgende Beispiel einer Befehlsdatei bezieht sich auf einige SONIX-Stücke, die auf der RPD-Diskette 107B zu finden sind:

```
D=CAM_107b:Scores    ; Aktuelles Verzeichnis setzen
I=/TVA_Instruments   ; Instrumentenverzeichnis setzen
V=-1                 ; Abspielen in Originallautstärke
;
C=2                  ; Nächstes Stück zweimal spielen
A="T.V.A (1)"        ; Stück laden und spielen, nächstes
                     ; Stück sofort laden
;
R=0                  ; Folgenden Block endlos wiederholen
  A="T.V.A (2)"      ; Dieses und nächstes Stück abwechseln
  A="T.V.A (4)"      ; spielen und laden
                     ; Blockende
```

Nach dem Abspeichern können Sie diese Befehlsdatei mit

```
Play [*] P=<Befehlsdatei>
```

 © 1990 technicSupport

abrufen. Zum Austesten längerer Befehlsdateien sollte PLAY ohne die *-Option gestartet werden. Dann läßt sich nämlich mit der Tastenkombination [CTRL]-[E] die Abarbeitung eines Stückes abbrechen und zum nächsten Titel springen.

Starten von der Workbench

Bevor Sie Ihre Musikstücke von der Workbench aus abspielen können, müssen Sie mit Hilfe des CLI einige Vorbereitungen treffen. Kopieren Sie PLAY in das Basisverzeichnis der Diskette, die Ihre Musikdaten enthält. Wenn sich Ihre Titel in einem gesonderten Verzeichnis befinden, müssen Sie das „Scores"-Icon von RPD 107 A in das Verzeichnis kopieren, in dem Ihr Titelverzeichnis liegt, und es mit dem Namen des Titelverzeichnisses versehen, z.B.:

```
cd <Verzeichnis, in dem Ihr Titelverzeichnis liegt>
copy CAM_107a:Scores.info <Titelverzeichnis>.info
```

Nun müssen Sie jeden Titel mit einem Icon versehen. Verwenden Sie hierzu eine der entsprechenden .info-Dateien von RPD 107A, z.B:

```
cd <Zieldiskette>:<Titelverzeichnis>
copy CAM_107a:Scores/YTV.smus.info <Titel>.info
```

Diese Icons müssen Sie noch mit der Information versorgen, wo die Instrumente zu finden sind. Wechseln Sie auf die Workbench, aktivieren Sie das jeweilige Icon und wählen Sie Info aus dem Workbench-Menü. Geben Sie im TOOL-TYPES-Eingabefeld den Pfad und den Namen des Instrumentenverzeichnisses ein, ausgehend von dem Verzeichnis, in dem sich das Icon befindet, etwa:

```
I=/<Instrumentenverzeichnis>
```

Nachdem Sie das Info-Fenster durch Anklicken des [Save]-Buttons verlassen haben, können Sie PLAY mit einem Doppelklick auf das Icon starten.

Soll PLAY von der Workbench aus veranlaßt werden, eine Befehlsdatei abzuarbeiten, macht sich das Fehlen des „Script"-Icons unangenehm bemerkbar, da das „Basteln" eines solchen Icons doch mit

einiger Arbeit verbunden ist. Zweckmäßigerweise verwendet man hierzu das Tool `Xicon` (z.B. auf Fishdisk 102), welches den Aufruf von CLI-Kommandos von der Workbench-Ebene aus möglich macht. `Xicon` ist an anderer Stelle in diesem Buch ausführlich beschrieben. Dort finden Sie auch konkrete Hinweise für die Herstellung eines „Script"-Icons.

PROBLEME

Schon wenn ein einziges Instrument nicht gefunden wird, bricht `PLAY` seine Arbeit mit der Fehlermeldung

```
PLAY can't load: <Titel>
```

ab. Obwohl hier immer der Titel angegeben wird, liegt der Fehler erfahrungsgemäß bei den Instrumenten. Überprüfen Sie in solchem Fall nochmals alle Pfadangaben und sehen Sie nach ob alle benötigten Instrumente auch wirklich im geforderten Verzeichnis liegen. Wenn hier alles in Ordnung ist, haben Sie möglicherweise nicht genug freien Speicher zur Verfügung. Dieser Fall kann besonders leicht bei Verwendung digitalisierter Instrumente eintreten.

Wenn `PLAY` im Multitaskingbetrieb ein Skript abarbeitet, während gleichzeitig z.B. eine Slide-Show läuft, dann sollten sich Musik- und Grafikdaten in zwei getrennten Laufwerken befinden, da sonst der Lesekopf ständig zwischen mehreren Dateien hin- und herwandern muß. Die Ladezeiten würden hierdurch eine unerträgliche Länge erreichen.

HERKUNFT

`PLAY` findet sich auf der RPD-Diskette 107A. Das Programm ist Shareware und sollte mit $10 honoriert werden. Autor ist:

```
Mark Riley - Play
P.O. Box 234
Simi Valley, CA 93062
```

 © 1990 technicSupport

NAME

`PrintPop` — Druckerkommandos von der Workbench

AUFRUF

```
run PrintPop >NIL:
```

ZWECK

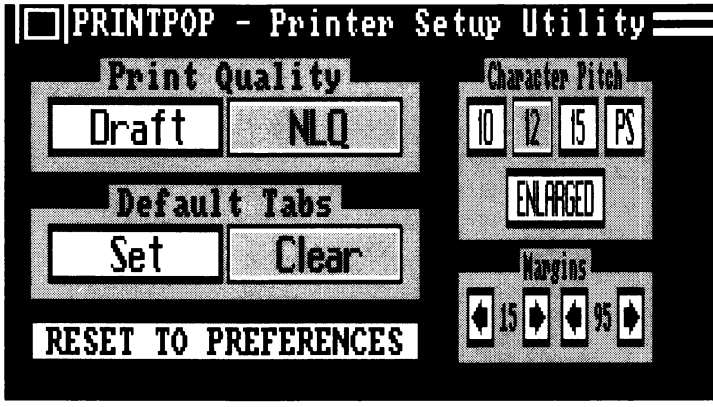
`PrintPop` gestattet es, auf einfache Weise einige wichtige Druckereinstellungen zu ändern, ohne dazu die `Preferences` aufrufen zu müssen.

BESCHREIBUNG

`PrintPop` ist so nützlich, daß man es in das c-Directory seiner Workbench-Diskette kopieren sollte. Fügt man den Aufruf in die `startup-sequence` ein, so steht `PrintPop` jederzeit im Speicher bereit. Mit der Tastenkombination [Amiga]-[F1] wird `PrintPop` zum Leben erweckt. Es öffnet sich ein kleines Fenster, in dem durch einfaches Anklicken von Gadgets folgende Parameter bestimmt werden können:

[NLQ]	Schönschrift
[DRAFT]	EDV-Qualität
[SET]	Tabulatoren werden berücksichtigt.
[CLEAR]	Tabulatoren bleiben unberücksichtigt.
[10]	Schriftart=PICA
[12]	Schriftart=ELITE
[15]	Schriftart=FINE
[PS]	Proportionalschrift
[ENLARGED]	Breitschrift
[⇐]	Der Wert für die linke oder rechte Randmarkierung wird um 1 verringert.

© 1990 technicSupport



PrintPop: Druckerhilfe für die Workbench

[⇒] Der Wert für die linke oder rechte Randmarkierung wird um 1 erhöht.

[RESET TO PREFERENCES]
Die in den `Preferences` festgelegte Einstellung wird aktiviert.

Nach dem Anklicken des [•] -Gadgets schließt sich das Fenster, und die erforderlichen Steuersequenzen werden an den Drucker geschickt. In den `Preferences` muß dazu der richtige Drucker eingestellt sein, auch haben die Steuersequenzen nur dann eine Wirkung, wenn der angeschlossene Drucker auch über die jeweils aktivierte Funktion verfügt. Beim ersten Aufruf von `PrintPop` muß das Programm übrigens Zugriff auf die Workbench haben, um den Druckertreiber laden zu können.

Wenn `PrintPop` erscheint, wird jedesmal der Workbench-Screen in den Vordergrund geholt. Befand man sich beim Aufruf auf einem anderen Screen, so kann dieser mit der Tastenkombination linke

 © 1990 technicSupport

[Amiga] - [M] wieder zurückgeholt werden. Wird das PrintPopFenster geschlossen, während andere Daten zum Drucker geschickt werden, so erscheint ein Fehlerrequester. Mit [Proceed] kann man seine Arbeit fortsetzen. Die Steuercodes werden an den Drucker geschickt, jedoch erst ausgewertet, wenn die noch im Druckerpuffer befindlichen Daten abgearbeitet sind.

PROBLEME

Wird PrintPop nach dem Starten des Systems benutzt, bevor irgendetwas ausgedruckt wurde, dann werden seine Einstellungen beim ersten Drucken von den Preferences überschrieben. PrintPop hat also die Aufgabe, die Druckerparameter zu ändern, nicht aber, sie zu aktivieren.

PrintPop erscheint mit eigenen Voreinstellungen, die nicht jedermanns Wünschen entsprechen dürften. Leider lassen sich diese Voreinstellungen nur im mitgelieferten C-Code ändern.

HERKUNFT

PrintPop findet man auf der Panorama 23c. Autor ist:

Robbie J. Akins
BEC
BCNZ
P.O.Box 98
Wellington
NEW ZEALAND.

VERWEISE

Purty (Fishdisk 066)

PointerAnimator

Der Mauszeiger wird animiert

AUFRUF

`PointerAnimator` ist ein Programmpaket, das aus zwei Einzelprogrammen besteht: `pa` zum Aktivieren eines animierten Mauszeigers und `merge` zum Erzeugen einer Animationsdatei für den Zeiger. Für die Erarbeitungs- und Testphase bietet sich der Aufruf aus dem CLI an. Dieser lautet:

`pa` ⟨*Animationsdatei*⟩ [-s⟨*n*⟩] [-h⟨*n*⟩] [-x⟨*n*⟩] [-y⟨*n*⟩]

bzw.

`merge` ⟨*Animationsdatei*⟩ [t⟨*Textdatei*⟩] {⟨*Bilddatei*⟩}

Mitgeliefert wird ebenfalls ein Lo-Res-IFF-Bild namens `DesignGrid` mit einem Raster, welches das Erstellen der Bilddateien erleichtern kann.

OPTIONEN

`Animationsdatei`
: Der Name der Datei, in der eine mit `merge` erzeugte Animation gespeichert wird. Mit dem Kommando `pa` wird eine solche Animation aktiviert.

`-s<n>`
: Diese Option legt die Abspielgeschwindigkeit in 50/n Bildern pro Sekunde fest. Voreingestellt ist ein Wert von 5, was genau 10 Bildern pro Sekunde entspricht. Gibt man hier den Wert 1 an, so erhält man die maximal mögliche Abspielgeschwindigkeit von 50 Bildern pro Sekunde.

`-h<n>`
: Hier wird die Höhe des Mauszeigers (und damit auch der Einzelbilder) in Bildschirmpunkten (*Pixels*) festgelegt. Erlaubt sind Werte von 1 bis 255. Stimmt diese Einstellung nicht mit der tatsächlichen Höhe der Einzelbilder überein, so erhält man ein laufendes Bild, wie man es von schlecht justierten Fernsehgeräten kennt.

© 1990 technicSupport

`-x<n>` Dieser Wert bestimmt den waagerechten Abstand des sogenannten *hot spot* zur linken oberen Ecke der Einzelbilder und ist auf Null voreingestellt. Als *hot spot* (heißer Fleck) wird der Punkt bezeichnet, der — unabhängig von der Gestalt des Mauszeigers — seine tatsächliche Position bestimmt.

`-y<n>` Dieser Wert legt den senkrechten Abstand des *hot spot* zur linken oberen Bildschirmecke fest. Auch hier besitzt die Voreinstellung den Wert Null.

`Bilddatei`
Der Name einer IFF-Datei, die aus mehreren untereinanderliegenden Einzelbildern besteht. Solche Dateien lassen sich mit IFF-Malprogrammen wie `DPaint` erstellen und als *Brush* (Pinsel) abspeichern. Die genaue Gestalt dieser Dateien wird weiter unten behandelt.

`Textdatei`
Wird dem Programm `merge` mit dieser Option eine Textdatei übergeben, so erscheint beim späteren Aufruf des Mauszeigers mit `pa` zunächst ein Requester, der den Text dieser Datei anzeigt. Erst nach dem Wegklicken dieses Requesters wird der neue Zeiger aktiviert. Auf diese Weise lassen sich in einer Animationsdatei Hinweise auf den Urheber des Mauszeigers unterbringen.

ZWECK

Mit `PointerAnimator` lassen sich einfache bis sehr komplizierte bewegte Mauszeiger erstellen. Tanzende, Grimassen schneidende oder rotierende Mauszeiger eignen sich zwar nicht sonderlich für ernsthafte Anwendungen, können jedoch auf Demo-Disketten, Spielen oder anderen unterhaltsamen Anwendungen für besondere Effekte sorgen.

BESCHREIBUNG

Manipulationen am Mauszeiger ließen sich bisher auch innerhalb des `Pointer`-Fensters der Preferences durchführen, sie beschränk-

ten sich allerdings darauf, dem Mauszeiger eine neue, jedoch konstante Form zu geben. Mauszeiger, die ständig ihre Form oder Farbe ändern, lassen sich auf diesem Wege nicht herstellen, dabei können gerade animierte Pointer nicht nur unterhaltsam, sondern auch nützlich sein, da sie sich besser vom Bildschirmhintergrund abheben und — bei geschicktem Einsatz — den Benutzer auf bestimmte Ereignisse aufmerksam machen können. Um einen Eindruck vom Unterhaltungswert solcher Pointer zu bekommen, genügt es, einige der auf der Fishdisk mitgelieferten Beispiele anzuklicken. Mit dem Programm `KillPointer` läßt sich der gewohnte Mauszeiger jederzeit wiederherstellen.

An diesen Beispielen läßt sich auch das Programm **pa** zusammen mit seinen Optionen erproben. Nachdem man mit dem Befehl

```
cd AmigaLibDisk170:PtrAnim
```

in das richtige Verzeichnis gewechselt ist, kann der Mauszeiger mit Namen `Shock` aufgerufen werden:

```
pa shock
```

Bei dieser einfachsten Form des Aufrufes werden die Voreinstellungen aller Parameter übernommen. Man erblickt einen Mauszeiger, an dem kleine Flammen emporzüngeln. Mit dem Aufruf

```
pa shock -s1
```

erhält man dasselbe Resultat, jedoch bewegen sich die Flammen jetzt mit fünffacher Geschwindigkeit. Man kann auch ein wenig mit der Höhe experimentieren: Die Befehle

```
pa shock -s1 -h32
pa shock -s1 -h37
```

lassen den Pointer in doppelter Ausfertigung oder in senkrechter Bewegung erscheinen. Mit dem Aufruf

```
pa shock -y100
```

wird der *hot spot* um 100 Bildschirmpunkte nach unten versetzt. Um etwa ein Gadget anzuklicken, muß man einen solchen Zeiger

 © 1990 technicSupport

100 Punkte oberhalb des Gadgets positionieren, was unpraktisch ist, die Wirkung dieser Option jedoch besonders drastisch zeigt.

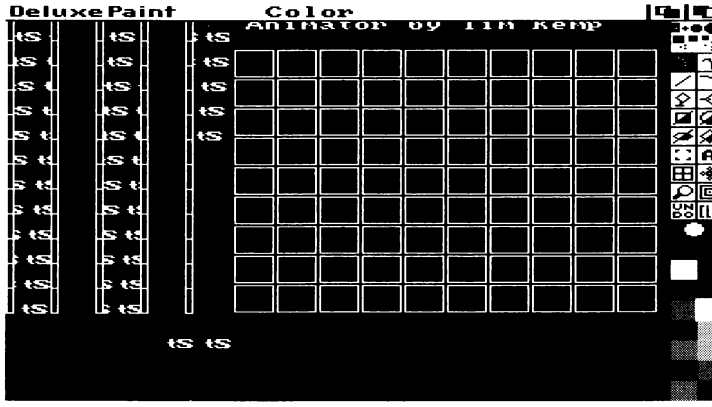
Das Erzeugen von neuen Mauszeigern

Eine Mauszeiger-Animation besteht aus einer Folge von Einzelbildern, die in schneller Folge nacheinander angezeigt werden. Für eigene Kreationen benötigt man daher ein Mal- oder Zeichenprogramm wie `DPaint`, mit dem sich Bildausschnitte in Form von sogenannten *Brushes* (Pinseln) abspeichern lassen. Da es auf detailgenaues Arbeiten ankommt, ist es vorteilhaft, wenn auch eine Koordinatenanzeige und eine Lupenfunktion vorhanden ist. Als Koordinatenanzeige kann auch das Programm `Surveyor` (s. Drittes PD-Buch, S. 248–250) von Fishdisk 151 herangezogen werden.

Für wirklich gute Kreationen benötigt man natürlich eine Idee, ein Konzept und das entsprechende künstlerische Talent. Da es das Schicksal in dieser Hinsicht mit dem Autor nicht besonders gut gemeint hat, begnügt er sich im folgenden mit einem zeichnerisch nicht sehr anspruchsvollen Beispiel. Das vorhandene Malprogramm wird gestartet, der LoRes-Modus mit 16 Farben (mehr sind nicht erlaubt) angewählt, dann wird das Hilfsraster-Bild `DesignGrid` eingeladen. Es kann sinnvoll sein, sich zuvor eine Arbeitsdiskette anzulegen und alle Programme aus der `PtrAnim`-Schublade auf diese Diskette zu kopieren.

```
format drive df<x>: name Ptr noicons
copy AmigaLibDisk170:PtrAnim Ptr:
```

Auf dieser Diskette sollten dann auch alle mit Hilfe des Malprogramms erzeugten Dateien abgelegt werden. Hat man das `DesignGrid` geladen, erblickt man in der rechten Hälfte ein Raster mit 90 Feldern der Größe 16×16 Pixel. In diesen Feldern können beliebige Einzelbilder editiert werden. Für andere Bildgrößen muß man sich ein eigenes Raster entwerfen. Auf der linken Seite gibt es einige senkrechte Hilfslinien, zwischen denen die fertigen Einzelbilder später untereinander angeordnet werden müssen. Eine solche Bildsäule wird dann als Brush herausgeschnitten und abgespeichert. In dieser Form können die Bildsäulen von `merge` verarbeitet werden.



Ein Raster erleichtert die Bildkomposition

Die Breite einer solchen Säule bestimmt auch die spätere Breite des Pointers: in unserem Fall 16 Bildpunkte, maximal möglich sind 32 Pixel. Die Höhe einer solchen Bildsäule muß ein exaktes Vielfaches der Einzelbildhöhe betragen, wie sie beim Aufruf des Pointers mit der `-s`-Option des `pa`-Programmes festgelegt wird.

Die Farbpalette kann den eigenen Wünschen angepaßt werden, nur die Hintergrundfarbe sollte unverändert bleiben, da nur sie auf der Workbench durchsichtig erscheint. Zu berücksichtigen ist ebenfalls, daß `pa` die Einzelbilder zyklisch abarbeitet, das letzte Bild der Animation also wieder in das erste Bild übergehen sollte.

In unserem Beispiel geht es darum, eine Laufschrift aus den Buchstaben „tS“ zu erzeugen, gewissermaßen ein einfaches Logo des Verlages technicSupport. Hierzu ist die rechte Seite des Rasters gar nicht nötig. Der Text „tS tS“ wird als Pinsel übernommen und, jeweils um einen Bildschirmpunkt versetzt, zwischen die senkrechten Hilfslinien positioniert. Der *hot spot* in der linken oberen Bildecke soll durch einen blinkenden Pfeil angezeigt werden. Dazu wird ein

 © 1990 technicSupport

kleiner Pfeil gemalt und in regelmäßigen Intervallen in die Einzelbilder eingefügt. Das Ergebnis läßt sich aus unserer Abbildung entnehmen: Es sind drei Bildsäulen (12 Bilder, 12 Bilder und einmal 5 Bilder), die als Brushes abgespeichert werden müssen. Alle Brushes müssen genau 16 Pixel breit sein, die ersten beiden 192 Pixel hoch, der dritte 80 Punkte. Unter welchen Namen die Pinsel abgespeichert werden, spielt keine Rolle. Heißen sie `col1`, `col2` und `col3`, dann lassen sie sich mit dem folgenden Kommando zu einer Animationsdatei zusammenfügen:

```
merge Zeiger col1 col2 col3
```

Getestet wird der neue Zeiger dann mit dem Befehl:

```
pa Zeiger
```

Möchte man eine Textdatei einfügen, so ist diese zunächst mit einem Editor zu erstellen (z. B. unter dem Namen `Hinweis`), bevor sie mit `merge` verarbeitet werden kann:

```
merge Zeiger -tHinweis col1 col2 col3
```

Möchte man einen Mauszeiger aus der `startup-sequence` heraus aufrufen, sollte man auf die Textoption verzichten, da sonst die Abarbeitung der Datei bis zum Wegklicken des Requesters unterbrochen wird.

Für den Aufruf aus der `Startup-Sequence` ist es erforderlich, das Programm `pa` und die Animationsdatei auf die Workbench-Diskette zu kopieren. Wer die Workbench 1.3 besitzt, sollte die Animationsdatei in die `Prefs`-Schublade kopieren, ansonsten bietet sich das `System`-Verzeichnis an. Das Programm `pa` ist am besten im `C:`-Verzeichnis aufgehoben:

```
copy Ptr:Zeiger sys:prefs
copy Ptr:pa c:
```

In die `startup-sequence` ist die Zeile

```
pa sys:prefs/Zeiger
```

einzufügen. Es besteht aber auch die Möglichkeit, den Mauszeiger über ein Icon zu starten, das vom Typ `Project` sein muß. Am besten

greift man auf das Icon eines der Beispiele zurück:

```
copy ptr:Shock.info sys:prefs/Zeiger.info
```

Von der Workbench aus läßt sich nun die `Prefs`- bzw. die `System`-Schublade öffnen und das `Zeiger`-Icon mit einem einfachen Mausklick aktivieren. Aus dem `Workbench`-Menü wählt man nun die `Info`-Funktion an. In das `Default Tool`-Eingabefeld trägt man Pfad und Namen des `pa`-Programmes ein, in unserem Fall lautet der Eintrag:

```
c:pa
```

Falls gewünscht, lassen sich im `Tool Types`-Feld auch die Optionen des `pa`-Kommandos berücksichtigen. Die erforderlichen Schlüsselwörter lauten: `SPEED` (Abspielgeschwindigkeit), `HEIGHT` (Bildhöhe), `X OFFSET` (horizontale Position des *hot spot*) sowie `Y OFFSET` (vertikale Position). Die Einträge könnten beispielsweise lauten:

```
SPEED=1
HEIGHT=37
Y OFFSET=100
```

Nach Verlassen des `Info`-Fensters mit [Save] steht der neue Mauszeiger auch über das Icon zur Verfügung. Die genaue Handhabung des `Info`-Befehls ist am Ende der Notepad-Beschreibung im Amiga-Benutzerhandbuch nachzulesen.

PROBLEME

Bisweilen können sich merkwürdige Resultate ergeben, wenn die Brushes versehentlich mit falschen Abmessungen abgespeichert wurden. Hier ist sorgfältiges Arbeiten gefragt.

HERKUNFT

Der `PointerAnimator` befindet sich auf der Fishdisk 170. Das Programmpaket ist Shareware und sollte mit $10–$20 honoriert werden. Die Adresse des Autors:

```
Tim Keep
P.O. Box 23101
Columbus, OH 43223
```

 © 1990 technicSupport

QView

Schneller Textbetrachter

AUFRUF

`QView` wird aus dem CLI heraus durch die Eingabe

`QView` ⟨*Textdatei*⟩

aufgerufen, wobei ⟨*Textdatei*⟩ für den Namen der anzuzeigenden Textdatei steht. Auf der Workbench kann `QView` mit Hilfe des `Info`-Menüs als Default Tool in das Icon einer Textdatei eingetragen werden.

ZWECK

`QView` ist ein sehr kurzes und sehr schnelles Programm zur Anzeige von Texten auf dem Bildschirm.

BESCHREIBUNG

Oftmals möchte man sich eine Textdatei schnell einmal auf dem Bildschirm ansehen, ohne sie zu bearbeiten oder auszudrucken. `QView` ist für solche Fälle hervorragend geeignet, weil es in Assembler geschrieben ist und sich auf die elementaren Funktionen eines Textanzeigers beschränkt. Auf Grund seiner Kürze (3372 Byte) ist es sehr schnell geladen. Auch der Textaufbau auf dem Bildschirm ist extrem schnell. Deutsche Sonderzeichen werden korrekt angezeigt, die PAL-Auflösung unterstützt. Gründe genug, sich `QView` einmal etwas näher anzusehen.

`QView` zeigt den gewünschten Text in einem eigenen Fenster; die Steuerung erfolgt ausschließlich über die Tastatur:

[Esc]	Das Programm wird verlassen.
[↑]	Eine Zeile aufwärts
[Shift] - [↑]	Eine Bildschirmseite aufwärts
[CTRL] - [↑]	Sprung zum Textanfang

© 1990 technicSupport

[Alt] - [↑]	Wie [CTRL] - [↑]
[↓]	Eine Zeile abwärts
[Shift] - [↓]	Eine Bildschirmseite abwärts
[CTRL] - [↓]	Sprung zum Textende
[Alt] - [↓]	Wie [CTRL] - [↓]
[→]	20 Spalten nach rechts
[←]	20 Spalten nach links
[CTRL] - [←]	Rücksprung an den linken Textrand
[Alt] - [←]	Wie [CTRL] - [←]
[S]	Am unteren Fensterrand öffnet sich ein Eingabefeld, in das ein Suchbegriff eingetragen wird. Nach Beendigung der Eingabe mit [Return] wird der Text in Vorwärtsrichtung nach dem Begriff durchsucht. Im Fenster wird der Textausschnitt von der ersten Zeile an gezeigt, die den Suchbegriff enthält.
[A]	Wiederholung des Suchvorgangs
[H]	Schaltet zwischen ASCII- und hexadezimaler Darstellung um, wie man sie vom CLI-Befehl `type opt h` kennt. Das bedeutet, daß man mit `QView` auch binäre Dateien problemlos betrachten kann.

HERKUNFT

Das hier in der Version 1.1 besprochene Programm befindet sich mit seinem Quellcode auf der Fishdisk 262. Es ist Shareware und sollte mit $10 honoriert werden. Die Anschrift des Autors:

```
Lyman R. Epp
10072 Wirt Plaza #15
Omaha, Nebraska 68134
```


© 1990 technicSupport

Ren

Auslagern von Zeichensatzdateien

AUFRUF

`ren` ⟨*Name*⟩`.font` ⟨*Neuer Pfad*⟩

OPTIONEN

`Name` Der Name einer der im `Fonts:`-Verzeichnis befindlichen Dateien, die die Informationen über den jeweiligen Zeichensatz enthalten.

`Neuer Pfad` Gibt den Pfad zu dem Verzeichnis an, in dem sich die zu dem Font gehörenden Zeichensatzdateien befinden. Dieser Pfad sollte immer mit dem Namen der jeweiligen Diskette beginnen.

ZWECK

`Ren` ermöglicht es, den Inhalt des `Fonts:`-Gerätes auf die `.font`-Dateien zu beschränken. Die platzraubenden Zeichensatzdateien können auf anderen Disketten untergebracht werden.

BESCHREIBUNG

Im Normalfall befinden sich alle nachladbaren Zeichensätze des Amiga im `Fonts`-Verzeichnis der Workbench-Diskette. Wenn man mit `Pagesetter`, `Notepad`, `Dpaint` oder vergleichbaren Programmen arbeitet, die solche Zeichensätze nutzen können, ist es oft wünschenswert, selbst erstellte oder aus der Public Domain übernommene Zeichensätze hinzuzufügen. Festplattenbesitzer haben hier keine Probleme, wer jedoch mit Disketten arbeiten muß, wird schnell feststellen, daß auf der Workbench-Diskette zuwenig Platz ist. Abhilfe schafft zunächst das Kopieren aller gewünschten Zeichensätze auf eine separate Diskette. Nach der Zuweisung

```
assign fonts: <Name der Font-Diskette>
```

können diese Zeichensätze dann von der neuen Diskette geladen werden. Es wird jedoch nicht sehr lange dauern, bis auch diese Diskette voll ist. Dann muß eine zweite Diskette her, wobei sich sofort das Problem stellt, daß jeweils nur eine dieser Disketten dem `Fonts:`-Gerät zugewiesen werden kann, die Zeichensätze aller selbstfabrizierter Font-Disketten also nie gleichzeitig verfügbar sind. Wenn man mit sehr vielen Zeichensätzen arbeiten möchte oder muß, wird das Hantieren mit diesen Disketten schnell zu einem Alptraum.

Daß dies nicht so sein muß, zeigt ein kleines Programm von STEVEN VERMEULEN. Betrachtet man ein typisches `Fonts`-Verzeichnis mit dem `List`-Kommando, so stellt man fest, daß zu jedem Zeichensatz eine Datei, die die Endung `.font` aufweist, und ein Unterverzeichnis gleichen Namens gehört:

```
ruby.font                    784 ----rwed 29-Jan-78 01:11:30
ruby                         Dir ----rwed 29-Jan-78 01:12:37
```

Die `.font`-Datei enthält die Information, in welchen Größen ein Zeichensatz vorhanden ist und wo er sich befindet. Betrachtet man eine solche Datei mit dem `Type`-Befehl

```
type fonts:ruby.font opt h
```

dann findet man gleich am Anfang der Datei den Eintrag:

```
0000: 0F000003 72756279 2F313200 00000000    ....ruby/12.....
```

Dies besagt, daß eine Datei 12, die den Ruby-Zeichensatz in der Größe 12 enthält, im Unterverzeichnis `Ruby` zu suchen ist. Existiert ein Zeichensatz in mehreren Größen, so enthält seine `.font`-Datei die entsprechende Anzahl solcher Einträge. Die Dateien, die die Gestalt der Zeichen festlegen, befinden sich also im `Ruby`-Unterverzeichnis, das sich ebenfalls mit dem `List`-Befehl betrachten läßt:

```
12                          4556 ----rwed 29-Jan-78 01:12:34
15                          5976 ----rwed 29-Jan-78 01:12:36
8                           3524 ----rwed 29-Jan-78 01:12:38
```

Von entscheidender Bedeutung ist der Umstand, daß die Datei `Ruby.font` mit 784 Byte im Vergleich zu den anderen drei Dateien sehr klein ist. Könnte man diese Dateien in anderen Verzeichnissen

 © 1990 technicSupport

unterbringen, würde das `Fonts:`-Verzeichnis lediglich aus den kurzen `.font`-Dateien bestehen, was eine enorme Platzersparnis bedeuten würde.

Genau dieser Effekt kann mit Hilfe des Programmes **Ren** erzielt werden. Das Betriebssystem orientiert sich nämlich bei der Suche nach den Zeichensatzdaten an den `.font`-Dateien, deren Einträge (im oberen Beispiel: `ruby/12`) sich mit dem Programm **Ren** ändern lassen.

Das Verfahren läßt sich am besten an einem Beispiel demonstrieren. Wir verwenden hier die Workbench 1.3; möchte man die Workbench 1.2 verwenden, so muß lediglich der Name an diese Diskette angepaßt werden. Wir booten Amiga mit einer Kopie der Workbench, begeben uns in das CLI und formatieren zunächst eine Leerdiskette, die später die Zeichensatzdateien aufnehmen soll:

```
format drive df1: name MeineFonts noicons
```

Wir kopieren das komplette `Fonts`-Verzeichnis der Workbench auf die neue Diskette und löschen auf dieser die dort nicht benötigten `.font`-Dateien:

```
copy fonts: MeineFonts: all
delete MeineFonts:#?.font
```

Auf der Workbench können hingegen alle Unterverzeichnisse des `font`-Directories gelöscht werden:

```
cd fonts:
delete ruby opal sapphire diamond emerald garnet topaz all
```

Nun sollte die RPD-Diskette in ein freies Laufwerk gelegt werden, um das Programm in das `c`-Verzeichnis der Workbench zu kopieren:

```
copy df1:ren c:renfont
```

Dabei erhält das Programm gleichzeitig den neuen Namen **Renfont**, was bei der Workbench 1.3 erforderlich ist, da das Kommando **Ren** schon vergeben ist. Es ist in der Datei `s:Shell-Startup` über den `alias`-Befehl als Variante des `rename`-Kommandos definiert, die eine

bedingte Verarbeitung von Jokerzeichen zuläßt. Der Aufruf `Ren` startet bei der Workbench 1.3 daher immer diese Variante des `rename`-Befehls.

Mit `Renfont` können wir nun alle `.font`-Dateien der Workbench so ändern, daß die Zeichensatz-Dateien auf der neuen Diskette gesucht werden:

```
cd fonts:
renfont ruby.font MeineFonts:ruby
renfont opal.font MeineFonts:opal
renfont sapphire.font MeineFonts:sapphire
renfont diamond.font MeineFonts:diamond
renfont emerald.font MeineFonts:emerald
renfont garnet.font MeineFonts:garnet
renfont topaz.font MeineFonts:topaz
```

Betrachtet man die geänderte Datei `ruby.font` mit dem `type opt h`-Befehl, stellt man fest, daß der Eintrag nun auf das neu eingerichtete Verzeichnis weist:

```
0000: 0F000003 6D65696E 65666F6E 74733A72    ....meinefonts:r
0010: 7562792F 31320000 00000000 00000000    uby/12..........
```

Um weitere Fonts hinzuzufügen, muß man deren `.font`-Dateien in das `Fonts:`-Verzeichnis der Workbench kopieren, die korrespondierenden Unterverzeichnisse jedoch auf die `MeineFonts`-Diskette. Für die Zeichensätze der `Extras1.3`-Diskette funktioniert das folgendermaßen:

```
copy Extras1.3:fonts/#?.font fonts:
copy Extras1.3:fonts MeineFonts: all
delete MeineFonts:#?.font
cd fonts:
renfont times.font MeineFonts:times
renfont courier.font MeineFonts:courier
renfont helvetica.font MeineFonts:helvetica
```

Obwohl sich mit dieser Workbench-Diskette jetzt mehr Zeichensätze ansprechen lassen als vorher, verbleibt auf ihr noch eine ganze Menge Platz für weitere Fonts.

 © 1990 technicSupport

Zum Ausprobieren kann man zum Beispiel das Notepad aufrufen. Im Auswahlmenü werden alle Zeichensätze angezeigt, die mit einer `.font`-Datei auf der Workbench vertreten sind. Erst wenn einer dieser Zeichensätze tatsächlich benötigt, d. h. angewählt wird, wird er von der `MeineFonts`-Diskette nachgeladen. Befindet diese sich nicht in einem der Laufwerke, wird sie von einem Requester angefordert.

Ist die Workbench doch einmal voll geworden, so lassen sich die mit `Ren` geänderten `.font`-Dateien ebenfalls auf einer separaten Diskette unterbringen, die dann mit dem `assign`-Befehl als `Fonts:`-Gerät deklariert werden muß. Auf einer solchen Diskette dürften sich dann an die 800 `.font`-Dateien unterbringen lassen, d. h. man kann ca. 800 verschiedene — unterschiedliche Größen nicht mitgerechnet! — Zeichensätze ansprechen, ohne ständig mit dem `assign`-Befehl operieren zu müssen. Die eigentlichen Zeichensätze sind dann natürlich über viele weitere Disketten verteilt, die sich in ihrem Namen voneinander unterscheiden und auch griffbereit sein müssen, da sie ja bei Bedarf angefordert werden.

PROBLEME

Bei der Festlegung des neuen Pfades ist unbedingt darauf zu achten, daß dieser Pfad mit dem Namen der Diskette beginnt. Laufwerksangaben wie `df1:` dürfen keinesfalls verwendet werden, weil sie in dieser Form in die `.font`-Datei geschrieben werden. In diesem Falle würde im Bedarfsfall nicht die Diskette mit dem betreffenden Namen angefordert, sondern es würde versucht, die Daten von der gerade in `df1:` liegenden Diskette zu lesen, wo sich höchstwahrscheinlich aber eine falsche Diskette befinden wird.

Verwaltet man seine Zeichensätze mit `Ren`, dann darf das Programm `FontFixer` nicht mehr auf diese Dateien angesetzt werden.

HERKUNFT

Das Programm `Ren` befindet sich mit seinem Quellcode in C auf der RPD 162. Autor ist:

`Steven Vermeulen, 3635 Utah Dr. N.W.`
`Calgary, Alberta, CANADA, T2N 4A6`
`(403) 282-7990`

RunBack

Komfortableres Starten von Hintergrundtasks

AUFRUF

Das Programm wird im CLI durch Eingabe von

`RunBack` [-⟨*Verzögerungsdauer*⟩] [-s⟨*Stackgröße*⟩] ⟨*Dateiname*⟩

aufgerufen.

OPTIONEN

`Verzögerungsdauer`
: Mit einer Zahl von 0 bis 9 läßt sich eine Verzögerungsdauer in Sekunden angeben, die das Programm verstreichen läßt, ehe ein weiterer Befehl abgearbeitet werden kann. Wird `RunBack` aus einer Batchdatei heraus aufgerufen, so verhindert diese Option, daß das Laufwerk auf zwei Dateien gleichzeitig zugreifen muß. Dies würde den Lesekopf nämlich zu häufigem Spurwechsel zwingen und die Ladezeit beträchtlich erhöhen.

`Stackgröße`
: Durch Angabe eines Zahlenwertes läßt sich die Stackgröße in Bytes für den Hintergrundtask festlegen. Entfällt diese Amgabe, wird die Stackgröße des gerade aktiven CLI-Tasks (normalerweise 4000) übernommen.

ZWECK

`RunBack` startet Hintergrundtasks, die nicht an ein CLI-Fenster gebunden sind.

BESCHREIBUNG

Auf einem multitaskingfähigen Computer wie `Amiga` laufen viele Prozesse als Hintergrundtask ab. Solche Prozesse lassen sich nicht

 © 1990 technicSupport

interaktiv steuern, sie laufen — oftmals unbemerkt — neben weiteren Programmen ab. Grundsätzlich kann jeder AmigaDOS-Befehl als Hintergrundtask gestartet werden, indem ihm der Befehl run vorangestellt wird, z. B.

```
run type irgendeinentext to prt:
```

Diese Methode bringt jedoch einen großen Nachteil mit sich: Das Fenster, von dem aus der Aufruf erfolgte, kann nicht mehr geschlossen werden, ehe der Hintergrundprozeß beendet worden ist. Diesen Effekt können Sie gut beobachten, wenn Sie schnell hintereinander folgende Zeilen eingeben:

```
run wait 30 secs
endcli
```

Das Fenster schließt sich nicht sofort nach Eingabe des Befehls, sondern erst ca. 30 Sekunden später. Auch mit Hilfe des NewCLI-Befehls läßt sich dieses Problem lediglich auf ein anderes Fenster verlagern. Besonders unpraktisch ist dieser Umstand, wenn ein Hintergrundtask über längere Zeit aktiv bleiben soll, da man ständig ein im Grunde überflüssiges Fenster mit sich „herumschleppen" muß. Genau hier sorgt RunBack für Abhilfe, da die so aufgerufenen Tasks nicht mehr an ein Fenster gekoppelt sind. Sofort nach der Eingabe von

```
RunBack c/wait 30 secs
endcli
```

verschwindet das Fenster, obwohl der Hintergrundtask noch läuft. Besonders geeignet ist RunBack daher für die Verwendung innerhalb der Startup-Sequence.

BEISPIEL

Ein Beispiel für die Verwendung von RunBack befindet sich im Anwendungsbeispiel über die Herstellung einer Arbeitsworkbench mit nützlichen PD-Utilities.

PROBLEME

Dem auszuführenden Befehl muß der komplette Pfad vorangestellt werden, andernfalls kann `RunBack` ihn nicht finden. Über `RunBack` dürfen nur ausführbare Programmdateien gestartet werden. Beim Aufruf anderer Dateien — was aber ohnehin keinen Sinn macht — stürzt das Programm ab.

Das Problem der Pfadangabe läßt sich beseitigen, wenn man die neueren `RunBack`-Versionen von Fishdisk 152 bzw. 214 verwendet. Diese Versionen berücksichtigen den Suchpfad automatisch. Wer allerdings mit der ARP-Shell arbeitet (vgl. „Workbench 1.3 für Individualisten") und `RunBack` resident laden möchte, kann dies nur mit der alten, nicht aber mit den neueren Versionen bewerkstelligen.

BEMERKUNGEN

Das Programm ist in Assembler geschrieben. Der Quellcode wird mitgeliefert.

HERKUNFT

`RunBack` findet sich auf Fishdisk 065. Autor ist CHARLIE HEATH.

VERWEISE

`RunBackGround` (Fishdisk 073), `RunBack` (Fishdisk 152 und 214)

© 1990 technicSupport

SafeBoot

Archiv für Bootblöcke

AUFRUF

`SafeBoot` kann sowohl im CLI als auch durch Doppelklick auf der Workbench gestartet werden.

ZWECK

Mit `SafeBoot` lassen sich Bootblöcke von Originaldisketten komfortabel speichern und nach einem Virusbefall des Originals ebenso komfortabel wieder zurückkopieren, um das Virus zu löschen und die Lauffähigkeit wiederherzustellen. `SafeBoot` ist wesentlich einfacher zu bedienen als das im Zweiten Amiga Public Domain Buch beschriebene Programm `BootBack`, das sich ausschließlich von CLI aus steuern ließ.

BESCHREIBUNG

Boot-Block-Viren haben für den Besitzer von Originalprogrammen nach wie vor ihren Schrecken nicht verloren. Kann ein Bootblock-Virus auf einer PD-Disk problemlos wieder gelöscht werden, so ist dies auf vielen Originaldisketten nicht möglich, da diese eigene Lade-Routinen, Kopierschutzabfragen oder ähnliches im Bootblock abgelegt haben und mit einem Virus im Bootblock nicht mehr funktionsfähig sind. Bei diesen Disketten nützt auch das Überschreiben des Virus mit einem Standardbootblock nichts mehr, da die spezifischen Routinen im ehemaligen Bootblock nicht wiederhergestellt werden können.

`SafeBoot` bietet einen Ausweg aus dieser Misere, indem es zwar nicht vor Virenbefall schützt (was auch nicht möglich ist, da jeder Schutz irgendwann von einem geschickten Programmierer umgangen werden wird und den Benutzer so nur in trügerischer Sicherheit wiegt), es aber ermöglicht, vom Original-Bootblock eine Sicherheitskopie anzufertigen und im Falle eines Falles einfach zurückzukopieren. Dies ist problemlos möglich, da der Bootblock nicht kopiergeschützt werden kann: Schließlich muß ihn der Amiga beim Booten

lesen können, weshalb Fremdformate oder ähnliche Kopierschutz-Mechanismen für diese eine Spur ausscheiden.

Anm. d. Hrsg.: Der obenstehende Absatz könnte so verstanden werden, daß Programme zur Viruserkennung nutzlos seien, was gewiß nicht in der Absicht des Autors liegt. Es gibt zwar keine absolute Sicherheit, aber die häufigsten Viren können mit einem regelmäßig aktualisierten Programm wie `VirusX` aufgespürt werden. Was nützt das schönste Bootblock-Archiv, wenn der eigene Amiga zum Infektionsherd für die Disketten aller Freunde geworden ist oder ein Virus sich an anderen Daten vergreift? Fazit: `SafeBoot` und `VirusX` sind keine Konkurrenz, sie ergänzen sich vielmehr.

`SafeBoot` selbst präsentiert sich dem Benutzer als kleines Fenster auf der Workbench, in dem zwischen Sichern und Restaurieren des Bootblocks gewählt und das zu verwendende Laufwerk angewählt werden kann.

Gestartet wird der Vorgang mit einem der beiden unter der Laufwerks-Zeile liegenden Gadgets [Volume Name] und [Custom Name]. Der Unterschied zwischen beiden liegt nur darin, daß bei [Custom Name] ein Dateiauswahl-Requester erscheint, in dem Verzeichnis und Dateiname, in/unter dem der Bootblock abgelegt werden soll, noch ausgewählt werden können, während der Bootblock beim Klick auf [Volume Name] unter dem Namen der entsprechenden Diskette im selben Verzeichnis wie `SafeBoot` abgelegt wird.

Letzteres ist insbesondere in Verbindung mit einer Sicherheitsabfrage von `SafeBoot` interessant: Stimmt der Name der Diskette, auf die der Bootblock zurückgeschrieben werden soll, nicht mit dem Namen überein, unter dem der entsprechende Bootblock abgelegt wurde, so macht `SafeBoot` den Benutzer mit einem Requester darauf aufmerksam und verhindert so, daß der Bootblock einer falschen Diskette überschrieben wird. Vollziehen wir das Ganze anhand eines Beispiels nach:

In Laufwerk 0 liegt die `Workbench 1.3D`, von der auch gebootet wurde, in Laufwerk 1 befindet sich die Fishdisk 175 (bzw. eine Kopie dieser Disk), deren Schreibschutz nicht gesetzt ist. Der Bootblock der Original-Workbench soll für alle Fälle gespeichert werden.

 © 1990 technicSupport

Dazu wird `SafeBoot` gestartet, die Gadgets [Save] und [DF0:] werden angeklickt und der Sicherungsvorgang durch einen Klick auf [Volume Name] gestartet. `SafeBoot` legt den Bootblock der Workbench nun unter dem Dateinamen Workbench 1.3D ab. Erleidet die Workbench nun irgendeinen Schaden in Form einer Infektion durch ein Bootblock-Virus, so wiederholt man den obigen Vorgang mit dem Unterschied, daß nun [Restore] anstelle von [Save] angeklickt werden muß.

Analog geht man bei einer Originaldiskette vor, die nach dem Start von `SafeBoot` natürlich erst ins entsprechende Laufwerk gelegt werden muß. Versucht man einen Bootblock unter einem Namen, der bereits existiert, zu speichern, erfolgt eine Sicherheitsabfrage.

Eventuell auftretende Fehler, ob beim Schreiben oder Lesen, werden hinter `Status:` erläutert. Bliebe als letztes noch das Gadget [Exit] zu erwähnen, welches, wie könnte es auch anders sein, das Programm beendet.

PROBLEME

Kann `SafeBoot` nicht auf die Datei `BootIcon` zugreifen, erhalten die abgelegten Bootblöcke kein Icon. Bei der Benutzung im CLI kann `BootIcon` dagegen entfallen.

HERKUNFT

`SafeBoot` stammt von der Fish Disk 175. Autor ist:

```
Mark Lanoux
4447 Markham Avenue
Jefferson, LA 70121 USA
```

Wer an dem Source Code von `SafeBoot` interessiert ist, kann diesen gegen Einsendung einer Leerdiskette samt Porto für die Rücksendung beim Autor beziehen.

VERWEISE

`BootBack` (RPD 112), `Virus-Control` (Fishdisk 211)

Setprefs

Laden einer beliebigen System-Configuration

AUFRUF

`setprefs` ⟨*Dateiname*⟩

ZWECK

`Setprefs` dient dazu, eine mit den `Preferences` erstellte Arbeitsumgebung zu aktivieren, ohne die `Preferences` aufrufen zu müssen. Auf diese Weise lassen sich mehrere Arbeitsumgebungen archivieren und sehr schnell gegeneinander austauschen. `Setprefs` dürfte insbesondere für Festplattenbesitzer interessant sein.

BESCHREIBUNG

In den `Preferences` lassen sich alle wichtigen Parameter für die Arbeitsumgebung einstellen: Drucker, Mausgeschwindigkeit, Farbeinstellung und vieles mehr. Die leichte Bedienbarkeit dieses Programms wird mit einer langen Ladezeit erkauft. Ein weiterer, entscheidener Nachteil liegt darin, daß die `Preferences` nicht in der Lage sind, mehr als eine Parameterdatei zu verwalten. Diese Parameterdatei trägt den Namen `system-configuration` und befindet sich im `devs`-Verzeichnis der Workbench-Diskette.

Als Folge dieser Beschränkung muß man für jede Änderung der Arbeitsumgebung die `Preferences` laden und die Änderungen per Hand vornehmen. Speichert man diese Änderungen ab, geht die vorherige Einstellung verloren. Wechselt man beispielsweise oft zwischen Schwarz-Weiß- und Graustufen-Druck, zwischen NLQ und EDV-Qualität, dann geht die Freude an der Arbeit rasch verloren. Festplatten-Besitzer sind hiervon besonders betroffen, da sie viele verschiedenartige Programme auf der Platte haben, beim Booten aber jedesmal die *system-configuration* von der Bootdiskette gelesen wird.

Mit `Setprefs` wird alles einfacher. Zuvor sind jedoch aus dem CLI heraus einige Vorbereitungen zu treffen. Am besten legt man sich

 © 1990 technicSupport

auf der Bootdiskette oder auf der Festplatte ein Verzeichnis an, in dem man die verschiedenen Parameterdateien unterbringt, zum Beispiel:

```
makedir devs:config
```

Nun sichert man seine Standard-Einstellung unter einem beliebigen Namen, indem man sie in das neu geschaffene Verzeichnis kopiert:

```
copy devs:system-configuration devs:config/standard
```

Nach dem Aufruf der **Preferences** durch

```
sys:preferences
```

kann man eine andere Einstellung vornehmen, z. B. für Ausdrucke in Graustufen, und die **Preferences** mit [Save] verlassen. Anschließend sichert man auch diese Datei:

```
copy devs:system-configuration devs:config/grau
```

Hat man alle benötigten Varianten erzeugt, kopiert man die Standard-Einstellung zurück, damit die **Preferences** wieder auf sie zugreifen können:

```
copy devs:config/standard devs:system-configuration
```

Den **Setprefs**-Befehl kopiert man in das `c:`-Verzeichnis, und ab sofort läßt sich jede vorhandene Arbeitsumgebung aktivieren, in dem ein Befehl der Art

```
setprefs devs:config/grau
```

eingegeben wird. Wichtig ist die genaue Pfadangabe für die jeweilige Parameterdatei. Solche Befehle lassen sich auch in eine Batchdatei schreiben, die mit einem Project-Icon versehen und über das Programm `Xicon` (s. Programmbeschreibung) von der Workbench aufgerufen werden kann. Dann müssen die Parameterdateien jedoch in ein Verzeichnis kopiert werden, welches mit einem Schubladen-Icon versehen ist.

Bei Festplatten ohne Autobootfähigkeit tritt ein weiteres Problem auf. Gewöhnlich findet sich in der `startup-sequence` eine Zeile der Art

```
assign devs: dh0:devs
```

die zur Folge hat, daß die `Preferences` nicht mehr auf das `devs`-Verzeichnis der Bootdiskette, sondern auf das der Festplatte zugreifen. Nimmt man in den `Preferences` irgendwelche Änderungen vor, werden diese daher auf der Festplatte abgelegt. Beim Bootvorgang wird jedoch immer die *system-configuration* der Bootdiskette ausgelesen, da dies geschieht bevor die `assign`-Zuweisung ausgeführt wird. Um die geänderte Einstellung zu aktivieren, muß man die `Preferences` aufrufen oder die neue *system-configuration* auf die Bootdiskette kopieren. Einfacher ist es, in die `startup-sequence` — nach dem erwähnten `assign`-Befehl — die Zeile

```
setprefs devs:system-configuration
```

einzufügen. So wird bei jedem Bootvorgang die Parameterdatei der Festplatte eingelesen.

HERKUNFT

Das Programm findet sich im `Preferences/config`-Verzeichnis der RPD 053. Autor ist:

`Willy G.J. Langeveld`

© 1990 technicSupport

ShoWiz

Zeigt Bilder an und ermöglicht Dia-Shows

AUFRUF

Über die Workbench als *Default Tool* im Info-Fenster eines Bildes oder einer Script-Datei, im CLI über den Befehl

`showiz` ⟨*Scriptdatei*⟩

oder

`showiz` {[+⟨*Sekunden*⟩] [−[a−z]] ⟨*Dateiname*| *Verzeichnisname*| *Textdatei*⟩} [!]

OPTIONEN

+⟨*Sekunden*⟩ Gibt die Zeit in Sekunden an, die vor dem Einladen des nächsten Bildes gewartet wird

Alle nachfolgenden Optionen beziehen sich auf die Effekte, die beim Einblenden des Bildes verwendet werden.

`-a` *Appear* — Einfaches Erscheinen des Bildes

`-b` *Blocks* — Bild erscheint in fünf Blöcken nacheinander.

`-c` *Checkerboard* — Bild erscheint mit einem Schachbrett-Effekt.

`-d` *Down* — Bild erscheint von oben nach unten.

`-e` *Enlarge* — Bild erscheint erst normal und wird dann auf das Vierfache vergrößert.

`-f` *Fade* — Bild blendet sich langsam ein.

`-g` *Grow* — Bild wächst von der Mitte her bis auf alle vier Ecken.

`-h` *Half* — Bild wächst von der Mitte her je zur Hälfte nach links und rechts.

`-i` *Invert* — Bild wird zuerst auf dem Kopf und danach richtig herum aufgebaut.

© 1990 technicSupport

`-j` *Join* — Gegenteil von `-h`: Das Bild „wächst“ in der Mitte zusammen.

`-k` Bild setzt sich aus 4 Blöcken zusammen

`-l` *Left* — Bild erscheint von links nach rechts.

`-m` *Mix* — Bild erscheint aus gemischten Blöcken, die sich danach von selbst ordnen.

`-n` Es wird eine Art Zebrastreifen in der Mitte von links nach rechts gezogen. Die fehlenden Teile erscheinen dann von der Mitte aus nach oben und unten

`-o` *Open* — Das Bild erscheint aus der Mitte heraus. Grundeinstellung, falls kein anderer Effekt angewählt wurde.

`-p` *Pull* — Das Bild wird von der linken Ecke her heruntergezogen.

`-q` *Quarter* — Das Bild erscheint erst normal und wird dann auf ein Viertel seiner Größe verkleinert.

`-r` *Right* — Bild erscheint von rechts nach links.

`-s` *Stripes* — Bild erscheint in Streifen.

`-t` *Trail* — Das Bild wird als eine Art Schlange aufgebaut, die sich von links oben über rechts, links, rechts und wieder links aufbaut.

`-u` *Up* — Bild erscheint von unten nach oben.

`-v` *reVersed* — Bild erscheint erst seitenverkehrt und wird dann richtig herum angezeigt.

`-w` *Weird* — Das Gegenteil von `-n`: Erst werden die Außenteile oben und unten aufgebaut, danach erscheint der Streifen in der Mitte.

`-x` *eXpand* — Bild vergößert sich von der Mitte aus auf Originalgröße.

`-y` Zeigt ein Bild erst verwischt an, danach erscheint das Original.

`-z` *Zebra* — Es erscheint eine Art Zebrastreifen von links nach rechts über die ganze Bildschirmhöhe. Die Zwischenräume werden dann von oben nach unten aufgefüllt.

© 1990 technicSupport

Einige dieser Effekte sind bei HAM-Bildern nicht zu verwenden. Dies merkt `ShoWiz` automatisch und benutzt stattdessen die Grundeinstellung `-o`.

! Erscheint als letzte Option in der Kommandozeile ein Ausrufezeichen, wiederholt sich die angegebene Kommandozeile ständig, bis über [Esc] abgebrochen wird.

ZWECK

Die Einsatzmöglichkeiten von `ShoWiz` erstrecken sich von der Verwendung als einfaches Anzeigeprogramm für Einzelbilder bis hin zum professionellen Einsatz von Dia-Shows zu Demonstrations- und Präsentationszwecken. Möglich wird dies durch viele zusätzliche Funktionen von `ShoWiz`, die weit über die von anderen Slideshow-Programmen gebotenen hinausgehen: Repräsentatives Anzeigen von Texten, Einbinden von Klängen sowie interaktive Benutzerabfragen.

BESCHREIBUNG

Um ein einzelnes Bild beispielsweise 10 Sekunden lang erscheinen zu lassen und mit dem Schachbrett-Effekt einzublenden, ist lediglich der Befehl

```
showiz +10 -c Bild.pic
```

nötig. Das geht auch mit mehreren Bildern:

```
showiz +10 -c Bild.pic -d Bild2.pic +5 -f Bild3.pic
```

Dabei bleiben die Optionen für Effekt und Zeit gültig, solange sie nicht explizit geändert werden. Sowohl das erste als auch das zweite Bild sind also jeweils für zehn, das letzte hingegen nur für fünf Sekunden zu sehen.

Um alle Bilder aus einem Verzeichnis betrachten zu können, braucht man jedoch nicht den Namen jedes einzelnen Bildes anzugeben, wenn statt einzelner Dateinamen ein Verzeichnisname angegeben wird. Alle Dateien, die keine Bilder sind, werden dabei automatisch ignoriert:

```
showiz +5 dh1:pictures !
```

zeigt zum Beispiel alle Bilder im Verzeichnis `dh1:pictures` an. Die Darstellungen sind für 5 Sekunden zu sehen und werden wiederholt, nachdem sie alle angezeigt wurden. Dabei ist zu beachten, daß `ShoWiz` die Bilder eines Verzeichnisses automatisch in der alphabetischen Reihenfolge anzeigt.

Die Bedienung von der Workbench aus ist noch einfacher. Einfach [SHIFT] -Taste drücken, festhalten und alle anzuzeigenden Bilder einmal anklicken. Als letztes folgt ein Doppelklick auf das `ShoWiz`-Icon, wodurch der Vorgang gestartet wird.

Möchte man ein Bild direkt durch einen Doppelklick auf das Icon — dieses muß vom Typ `Project` sein — anschauen, ist zuvor die `Info`-Funktion aus dem `Workbench`-Menü aufzurufen, um unter `DEFAULT TOOL` den Pfad und Namen des `ShoWiz` Programmes einzugeben. Befindet sich das Programm zum Beispiel im Hauptverzeichnis derselben Diskette, so muß hier

```
:Showiz
```

stehen. Um bestimmte Effekte und Darstellungszeiten für das Bild zu erreichen, trägt man direkt darunter in das `TOOL TYPES`-Feld für ein bestimmtes Muster

```
WIPE=-c (oder eine vergleichbare Option)
```

und für eine bestimmte Verweildauer

```
TIME=+10 (oder ein anderer Zahlenwert)
```

ein. Nun kann das Bild durch einen einfachen Doppelklick auf sein Icon mit den gewünschten Optionen gestartet werden.

Das Anzeigen von Texten

Für anspruchsvolle Präsentationen lassen sich mit `ShoWiz` auch Texte anzeigen — in Farbe und mit Schatten. Dazu muß lediglich eine Text-Datei erstellt werden, die in der ersten Zeile wie folgt aussieht:

```
Showiz.Textfile -- Background,Text,Shadow Colors-->*04-04-04*12-
                                                    08-02*00-00-00*
```

 © 1990 technicSupport

Die drei mal drei Zahlen am Ende der Zeile geben nacheinander die Rot-Grün-Blau-Werte der Farben für Hintergrund, Schrift und Schatten an. Diese Werte können zwischen 0 und 15 liegen. Möchte man keine Schatten, ist die Farbe für Schatten auf den für den Hintergrund gewählten Farbwert zu setzen.

Nun folgt in der zweiten Zeile eine Reihe von 79 Sternchen (*), die den Beginn des Textes anzeigen. Danach 21 Zeilen lang jeweils nur ein Sternchen an der ersten und eines an der 79. Position. Die letzte Zeile besteht nun wieder aus 79 Sternchen. Auf diese Weise hat man sich einen Rahmen geschaffen, der die nutzbare Fläche des Bildschirmes anzeigt. In diesen kann nun jeder beliebige Text geschrieben werden. Ruft man das Programm im CLI auf, überprüft `ShoWiz` zuerst, ob die erste Zeile mit der oben angegebenen identisch ist. Ist das der Fall, wird die Textdatei ausgewertet und in den entsprechenden Farben auf dem Bildschirm ausgegeben.

Das Erstellen von Scripts

Doch das waren nur die einfachsten Bedienmöglichkeiten des Programmes. Nun soll die integrierte Script-Sprache erläutert werden: Bei einem Script handelt es sich um eine Textdatei, deren Name `ShoWiz` mitgeteilt wird. Diese Datei enthält Kommandos zur Steuerung des Programmes, die vielfältigste Möglichkeiten eröffnen. Die Kommandos im einzelnen:

STEUERUNG DER BILDEINBLENDUNGEN

`Wipe` Wählt den Einblendeffekt für das nächste Bild an. Beispiel: `Wipe -g`

`Time` Wählt die Wartezeit bis zum Laden des nächsten Bildes. Beispiel: `Time +20`

`Show` Lädt das entsprechende Bild (oder die oben erwähnte Textdatei) ein und stellt es mit den vorher gewählten Optionen dar. Beispiel: `Show Bild1.pic`

Musik- und Klangeffekte

LoadSound Lädt einen digitalisierten Sound in den Speicher. Beispiel: `LoadSound Schrei.snd`

PlaySound Spielt den vorher geladenen Sound x-mal ab. Wird 0-mal angegeben, ertönt der Sound, bis das Programm abgebrochen wird. Beispiel: `PlaySound +5`

Channel Hiermit kann einer der beiden zur Verfügung stehenden Kanäle (0 und 1) für den Sound eingerichtet werden. Dies muß vor dem Laden eines Sounds geschehen! Beispiel: `Channel 0`

Quiet Stoppt die Sound-Wiedergabe auf dem angegebenen Kanal. Der Sound bleibt jedoch im Speicher und kann später wiederverwendet werden. Beispiel: `Quiet 0`

FreeChannel Wie `Quiet`, mit dem Unterschied, daß der Sound auch aus dem Speicher gelöscht wird. Das wiederum gibt meist recht viel des so oft benötigten CHIP-Speichers frei! Beispiel:
`FreeChannel 1`

Verschiedenes

WB-On Schaltet die Workbench-Farben wieder auf normale Werte um. Dies wird beim Beenden des Programmes auch automatisch getan. Normalerweise schaltet `ShoWiz` beim Abarbeiten einer Script-Datei alle Workbench-Farben auf schwarz um. Möchte man aus der Script-Datei ein anderes Programm starten, kann es nötig sein, die Farben kurzzeitig zu normalisieren.

Fade-WB Schaltet die Workbench-Farben wieder auf Schwarz.

Delay Hält die Ausführung des Scripts an dieser Stelle um die angegebene Anzahl von Sekunden an. Beispiel: `Delay +15`

© 1990 technicSupport

`Request` Läßt einen Requester mit den Gadgets [Yes] und [No] erscheinen. Wählt der Benutzer [No], wird die nächste Zeile im Script übergangen, ansonsten ausgeführt. Der Text nach Request, erscheint auf dem Bildschirm im Requester. Beispiel: `Request Soll ich die Demonstration abbrechen?`

`Alert` Verhält sich genau wie `Request`, mit dem Unterschied, daß kein Requester erscheint, sondern ein rot blinkender Alert, einer Guru-Meldung nicht unähnlich.

`Stop` Bricht die Ausführung des Scripts ab. Eigentlich nur sinnvoll direkt hinter einer `Alert`- oder `Request`-Anweisung.

`Repeat` Führt die gesamte Script-Datei noch einmal aus.

`Execute` Übergibt den Rest der Zeile als Kommando an das CLI, welches dieses dann ausführt. Im Klartext heißt das, daß hier alle nur erdenklichen Programme ausgeführt werden können. Beispiel:
`Execute dh0:util/showanim dh1:Amiga.anim`

Eine Script-Datei muß unbedingt mit der Zeile

```
*Showiz.Scriptfile*
```

beginnen. Alle folgenden Zeilen werden als Befehle im obigen Sinne interpretiert. Beim Auftreten eines Fehlers (falsche Schreibweise oder Datei nicht zu finden) übergeht das Programm diese Zeile einfach. Kommentare können der Script-Datei zugefügt werden, indem man als erstes Zeichen in einer neuen Zeile ein Semikolon tippt. Der Rest der Zeile wird dann als Kommentar angesehen und ignoriert.

PROBLEME

Leider gibt es keine Möglichkeit, die Beendigung des Programms auf andere Art und Weise geschehen zu lassen, als durch Herunterscrollen des Bildschirmes. Bei Ausführung des Programmes `Movie`, welches Animationen des Ray-Tracing-Programmes `Sculpt` zeigt, über den `Execute`-Befehl in einer Script-Datei stürzt der Rechner bisweilen ab. Eine Abhilfe ist nicht bekannt.

Natürlich werden viele Benutzer erst einmal über die Funktionsvielfalt des Programmes verblüfft sein. Doch gerade wenn man das Programm auf der TBAG-Disk 31 vor sich liegen hat, sind die beigefügten Beispiele sehr hilfreich. Dort findet man eine Demo-Scriptdatei und einige Bilder sowie Textdateien, die als Vorlage für die ersten eigenen Versuche herangezogen werden können.

HERKUNFT

Der Autor des hier in der Version 2.0 beschriebenen Programmes, J. L. WHITE, ist als Herausgeber der TBAG-Public-Domain-Reihe bekannt. Kaum verwunderlich also, daß man das Programm auf einer TBAG findet: auf Nr. 31. Der Autor kann unter der TBAG-Adresse

```
J. L. White
1304 Four Seasons Blvd.
Tampa, Fla. 33615
Telefon USA: (813) 971-4451
```

erreicht werden. `ShoWiz` 2.0 ist Shareware. Bei Übersendung von $10 wird man registrierter Benutzer mit Informationen über Updates. Für $15 bekommt man zusätzlich noch eine Diskette mit der neuesten Version des Programmes sowie vielen neuen Bildern.

VERWEISE

`ShoWizJr` (kleinere Version, ebenfalls auf TBAG 31), `SuperView` (TBAG 25), `HamGif` (ACS 150), `PhotonSlide` (RPD 125)

© 1990 technicSupport

startle

Von der Vision eines Hackers

AUFRUF

`startle` ⟨*Textdatei*⟩ [⟨*Font*⟩ einGröße] [⟨*rgb*⟩]

OPTIONEN

Textdatei	Name der Datei, die den anzuzeigenden Text enthält.
Font	Der Name des Fonts, in dem der Text angezeigt werden soll (z.B „Ruby“).
Größe	Die Größe des Fonts in Pixeln. Der Font muß sich in der gewünschten Größe im `Fonts:`-Directory befinden.
rgb	Drei Hexadezimalwerte (0-9,A,B,C,D,E,F), die die Intensität der jeweiligen Grundfarbe (rot, grün, blau) bestimmen. Diese Option darf nur angegeben werden, wenn auch alle anderen bestimmt werden — sonst Guru!

ZWECK

`startle` sorgt für eine effektive Präsentation von Texten. Der Text scrollt von unten nach oben in der gewählten Schriftart und Farbe über den Bildschirm. Dabei fliegen fortwährend Sterne auf den Betrachter zu, ein Effekt, wie man ihn von Raumschiff Enterprise (StarTrek) kennt. `startle` eignet sich somit besonders gut für Intros.

BEISPIEL

Um etwas über die Vision des Hackers zu erfahren, können Sie im CLI folgende Befehle eingeben:

```
cd df1:startle
startle hype
```

© 1990 technicSupport

Vorausgesetzt wird, daß sich die Panorama-Diskette im Laufwerk 1 befindet. Der Text der Datei „hype“ wird in der Standardeinstellung Topaz 8 und hellem Gelb präsentiert. Mit dem Befehl

```
startle hype sapphire 19 00f
```

wird die Schrift in blauer Farbe erscheinen. Abgebrochen wird `startle` mit einer beliebigen Taste, ansonsten wird der Text endlos wiederholt.

PROBLEME

Trotz unserer optimistischen Beschreibung führte die Angabe eines anderen Fonts so gut wie nie zum Erfolg. Die voreingestellte Schriftart blieb weiterhin gültig, nur einmal — bei Emerald 20 — wurde die andere Schrift akzeptiert.

HERKUNFT

Die besprochene Version 1.0 findet man auf der Panorama-Diskette 23c. Autor ist:

```
Hobie Orris, Toronto, CANADA
uucp: utzoo!sq!hobie
CIS: 73657,1573
```

VERWEISE

`startle` basiert auf dem Programm `Stars` von LEO L. SCHWAB. Dieses findet man auf Fishdisk 118.

© 1990 technicSupport

StoneAge

PD-Variante zu Emerald Mine

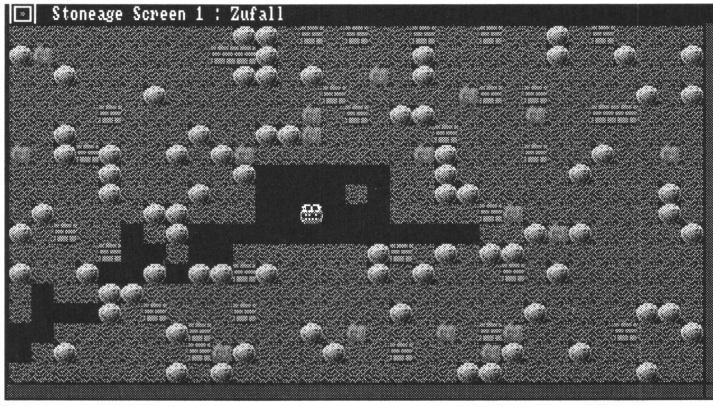
AUFRUF

`Stone-AgeV1.1` ist in Basic geschrieben, braucht aber die `Extras`-Diskette mit dem AmigaBasic-Interpreter nicht, da der Interpreter schon auf dieser selbststartenden Diskette der RPD-Serie vorhanden ist. Das Programm läßt sich von der Workbench aus mit einem Doppelklick auf sein Icon starten.

BESCHREIBUNG

`Emerald Mine` beziehungsweise dessen Vorläufer `Boulder Dash` gibt es jetzt als preiswertes Public Domain Programm. Zwar ist es in dem etwas langsamen Basic geschrieben, doch zum einen ist der Interpreter gleich auf der Diskette, zum anderen kann dadurch auch ein Neuling leicht den Programmcode verändern. Schaut man sich den Programmtext näher an, so sieht man, daß KLAUS KRAMER übersichtlich und sauber programmiert hat. Ein Hinweis hierfür ist, daß er eine Schnittstelle zur Erweiterung seines Programmes markiert hat. Auch kann der Anwender seiner Spielfigur — wir wollen sie, wie der Autor, Willi nennen — unendliche Leben oder Unverletzlichkeit schenken, aber bitte nur auf der Sicherungskopie!

Ist das Programm gestartet, gelangt man nach dem Lesen des Copyright-Vermerkes in den ersten Screen. Offensichtlich ist der Autor davon ausgegangen, daß alle Funktionen für dieses Spiel bekannt sind. Er erwähnt nur, daß die [Esc] -Taste für einen Selbstmord notwendig sei. Für den Fall, daß jemand die Spiele `Emerald Mine` oder `Boulder Dash` nicht kennen sollte, möchte ich hier kurz das Ziel dieser Spiele und die Vorgehensweise skizzieren: Mit einer Figur, hier Willi genannt, müssen möglichst schnell wertvolle Gegenstände wie Smaragde oder Diamanten eingesammelt werden. Dabei muß man auf herabfallende Steine, Sandsäcke oder auch sogenannte „Fresser“ achten. Hat man alle Diamanten eingesammelt, kommt man in den nächsten Level. Allerdings ist es äußerst schwierig, alle Diamanten einzusammeln, weil man zum einen durch herun-



Die PD-Version des Spielhallen-Klassikers Boulder Dash

terfallende Steine getötet, zum anderen durch eine vom Spiel vorgenommene geschickte Positionierung der Steine eingeschlossen werden kann. Einzelne Steine können durch „Wegdrücken" verschoben werden. In der Regel hat man mehrere Leben.

Bei `Stone-AgeV1.1` ist einiges einfacher: Es gibt 10 Leben, das Spiel hat sieben Level, es gibt keine Fresser, und die Zeit spielt keine Rolle. Es kommt also „nur" darauf an, die Diamanten — im Programmtext werden sie Kohl genannt — einzusammeln und aufzupassen, daß Willi nicht die Steine auf den Kopf fallen. Jedes Level hat einen Namen: Das erste heißt „Zufall", das zweite „Logik-Horror", das dritte „Jeder Schritt kann falsch sein", das vierte „Fallenstellers Nachtgebet", das fünfte „Unlösbar?", das sechste „Willis Brüder" und das siebte schließlich „Fred lebt! Lupo auch". Wie man an den Überschriften erkennt, hat der Autor jedes Level unter ein Thema gestellt — ich meine, daß diese Überschriften treffend sind.

 © 1990 technicSupport

Die sieben verschiedenen Spielstufen

Jetzt aber zu den einzelnen Leveln. Zum ersten Level braucht nicht viel gesagt werden: Die Diamanten, nein Kohlköpfe, können leicht eingesammelt werden. Aber Vorsicht vor den herabfallenden Steinen: Sie erschlagen Willi. Schnell sind die Kohlköpfe eingesammelt, und die Meldung über die gebrauchte Zeit erscheint. In diesem ersten Level sind die Kohlköpfe zufällig positioniert worden — hat der Zufall geschickt mitgespielt, dann kann man die Kohlköpfe schnell einsammeln.

Im nächsten Level wird es schon schwieriger, nicht umsonst heißt es „Logik-Horror". Nur ein Tip: Das Naheliegende ist oft nicht das Richtige. Aber nach einigen Versuchen werden Sie es schon hinkriegen. Wer es nicht schafft, dies meine ich besonders in Hinblick auf die höheren Level, der kann seinen Willi ganz einfach gegen Steinschlag immun machen (s. Anregungen).

Spannend wird es bei „Jeder Schritt kann falsch sein": Hoffentlich wird Willi nicht gleich zu Anfang erschlagen! Testen Sie die Wege aus, Sie haben ja noch einige Leben. Dem professionellen Emerald Mine-Spieler wird es keine Schwierigkeiten bereiten, diese Level zu meistern, für die anderen heißt es halt: Aller Anfang ist schwer.

Weiter geht es zu „Fallenstellers Nachtgebet": Dieses Level hat mir in der Tat einige Stoßseufzer abgerungen, bis ich den richtigen Dreh heraus hatte: Man muß sich beim letzten Stein einschließen lassen. In den drei letzten Leveln hatte ich beim sechsten — „Willis Brüder" — keine Schwierigkeiten: Willi muß hier halt lauter Willis einsammeln. Das letzte Level bringt noch einmal Spaß, aber warum es „Fred lebt! Lupo auch" heißt, ist mir unklar.

`Stone-AgeV1.1` macht mir immer wieder Freude: Zwar ist der Bildschirmaufbau zu Beginn eines neuen Levels etwas langsam, aber das ist halt typisch für AmigaBasic. Was mich an diesem Spiel fasziniert? Man kann es einfach seinen Wünschen anpassen, und es ist ein Spiel, wo Kombinations- und Merkfähigkeit gefragt sind. Wenn Sie alle Spiele bis einschließlich Level 7 geschafft haben, lassen Sie sich durch das letzte Bild überraschen — Es erscheinen lauter Willis, aber wie!

© 1990 technicSupport

Anregungen

Da der Programmcode des Spieles vorliegt, kann es von jedem, der ein wenig AmigaBasic versteht, leicht nach eigenen Vorstellungen abgeändert werden. Die Screens sind als Bilder nur mit ihren Datasegmenten abgelegt. Aber Sie können es ja einmal selbst probieren und eigene Bilder kreieren. Die Farben der Erde, d. h. des Hintergrundes, auf dem die Kohlköpfe und Steine plaziert sind, werden übrigens über eine Zufallszahl bestimmt. Noch etwas zum Ausprobieren: Die Prozedur, die man ausschalten sollte, um Willi für Trainingszwecke vorübergehend immun zu machen, heißt „itsovernowbabyblue". Die Numerierung der Bilder beginnt, nebenbei bemerkt, bei Null. Vielleicht läßt sich auch eine Prozedur zum Zerschießen von Steinen einfügen?

HERKUNFT

Die hier besprochene Version befindet sich auf der RPD 015. `StoneAgeV1.1` hat einen Copyright-Vermerk des Autors KLAUS KRAMER. In der Eingangsmeldung kann man „The Twilight & The Enemies" lesen. Eine Adresse ist leider nicht zu erfahren.

 © 1990 technicSupport

TurboBackup

Schnelles Diskettenkopieren

AUFRUF

`TurboBackup` kann von der Workbench aus durch einfaches Anklikken gestartet werden. Vom CLI aus erfolgt der Start mit

`TurboBackupv1.00` [⟨*Laufwerk*⟩ to ⟨*Laufwerk*⟩[, ⟨*Laufwerk*⟩ [, ⟨*Laufwerk*⟩]]] [-s]

OPTIONEN

`-s` Das Programm eröffnet nicht in einem Fenster, sondern in einem eigenen Bildschirm.

Durch die optionale Angabe der Laufwerke können Quell- und Ziellaufwerke bereits beim Aufruf bestimmt werden.

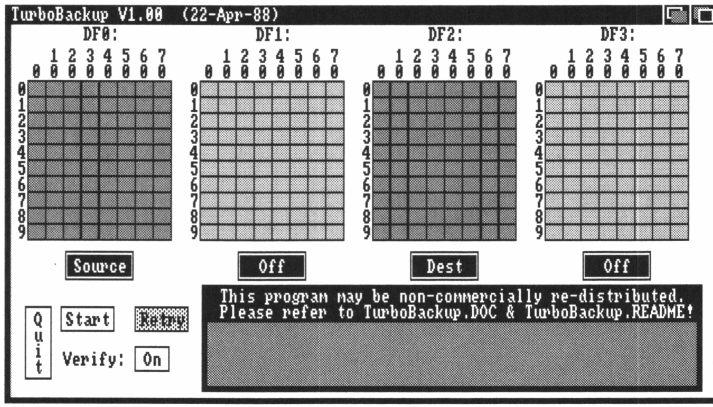
ZWECK

Der Standard-`DOS`-Befehl `diskcopy` prüft nach dem Kopieren nicht auf Richtigkeit der Aufzeichnung — es sei denn, man besitzt bereits die Workbench mit der Versionsnummer 34.28. `TurboBackup` erledigt das in der gleichen Zeit, wie sie sonst für das normale Kopieren ohne Prüfung benötigt wird!

Sie können mit `TurboBackup` auch von einem Laufwerk in einem Durchlauf auf mehrere Ziellaufwerke kopieren und so einen zusätzlichen Zeitvorteil erringen.

BESCHREIBUNG

Nach dem Aufruf des Programmes erscheint ein Fenster mit vier Rastern, welche die vier Laufwerke symbolisieren, die an den Amiga angeschlossen werden können. Falls Sie weniger Laufwerken betreiben, so erscheinen trotzdem vier Raster, Sie können dann aber nicht alle aktivieren.



TurboBackup ist eines der schnellsten Kopierprogramme

Um von einem Laufwerk auf ein anderes zu kopieren, drücken Sie das Kästchen unter dem Quellaufwerk einmal (es erscheint [Source]) und das Kästchen unter dem Ziellaufwerk zweimal (es erscheint [Dest]).

Sie können auch mit den Tasten [F1] , [F2] , [F3] und [F4] eines der Laufwerke 1 bis 4 als Quelle selektieren. Mit den Tasten [F6] , [F7] , [F8] und [F9] werden die Laufwerke 1 bis 4 als Ziellaufwerke gekennzeichnet. Mit [Shift] - [F1] , [Shift] - [F2] , [Shift] - [F3] und [Shift] - [F4] schließlich werden die Laufwerke wieder als inaktiv gekennzeichnet. Mit einem Klick auf [Start] beginnt das Kopieren. Stattdessen können Sie auch die Taste [S] drücken.

Ein kleiner Buchstabe, der sich in den dargestellten Rastern bewegt, symbolisiert den Ort des Schreib-/Lesekopfes. Durch die Buchstaben `r` (für *read*, Lesen), `w` (für *write*, Schreiben) und `v` (für *verify*, Prüfen) wird der Zustand des Laufwerkes angezeigt.

Falls Sie das Kopieren jetzt abbrechen möchten, so drücken Sie auf [Abort] oder die Taste [A] .

 © 1990 technicSupport

Wenn eine Spur beim ersten Versuch nicht gelesen werden kann, wiederholt TurboBackup diesen Versuch bis zu viermal. Eine Spur, die solche Probleme bereitet, wird durch einen orangefarbenen Kreis gekennzeichnet. Sie haben dann die Möglichkeit, durch [Retry] oder die Taste [R] einen neuen Versuch oder durch [Abort] den Abbruch zu verlangen. Das Programm läßt sich mit [Quit] oder der Tastenkombination [Shift]-[Q] beenden.

PROBLEME

Die Laufwerke werden vom Programm belegt, so daß andere Amiga-Programme nicht gleichzeitig auf sie zugreifen können.

Nach dem Kopieren von Disketten müssen alle (bis auf eine) entfernt werden. AmigaDOS würde sonst durch die gleiche Kennung der Disketten verwirrt werden. Als auf die Dauer doch mehr als unangenehm erweist sich die Tatsache, daß Amiga unter Umständen vollständig abstürzt, wenn sich zwei identische Disketten, wie sie TurboBackup produziert, in den Laufwerken befinden. Mir passierte dies einigemal. Man mag solch einen Vorfall als Bedienungsfehler, als Fehler von AmigaDOS oder als Fehler von TurboBackup ansehen. Tatsache ist aber, daß nur TurboBackup solche identischen Kopien herstellt, und der Autor aufgrund der dadurch auftretenden Systemhavarien die Arbeit mit diesem sonst so nützlichen Programm wieder aufgab.

Falls Sie in großem Umfang Disketten kopieren, kann Ihnen TurboBackup viel Zeit ersparen. Ich empfehle Ihnen aber, nicht vollständig und sofort umzusteigen, sondern in einem Testbetrieb erst einmal Erfahrungen mit dem Programm zu sammeln und den Übergang vorsichtig zu vollziehen. Das Ersetzen eines bewährten alten Programmes durch ein neueres, noch unbekanntes, ist stets mit dem Risiko unerwarteter Probleme verbunden.

HERKUNFT

Dieses Programm ist auf Fishdisk 139 zu finden. Es wurde von MARTIN KOPP und STEFFEN STEMPEL geschrieben. Ihre Adresse:

Martin Kopp, Frankenstr. 34, 6800 Mannheim 52

VC—AnalytiCalc

Spreadsheet-Programme für Tabellenkalkulationen

AUFRUF

Bei `VC` durch Anklicken des entsprechenden Icons; bei `AnalytiCalc` muß vor der Benutzung ein recht kompliziertes Installationsverfahren durchgeführt werden. Danach sollte eine eigene Workbench-Disk angefertigt werden, die das Programm aus der `Startup-Sequence` heraus aufruft. Ein Aufruf vom CLI aus kann durch Eingabe von

```
Analy
```

erfolgen.

ZWECK

Beide Programme unterstützen Kalkulationen und Planungen, denen viele verschiedene Faktoren und Größen zugrunde liegen können. Während `VC` recht einfach zu bedienen, in seiner Leistungsfähigkeit aber beschränkt ist, braucht sich `AnalytiCalc` in Sachen Funktionalität nicht hinter kommerziellen Programmen dieser Sparte zu verstecken.

BESCHREIBUNG

Haben Sie schon einmal eines jener Amiga-Spielprogramme benutzt, das eine sogenannte „Boss-Taste" hatte? Diese Tastenkombination dient Millionen von mehr oder weniger hart arbeitenden Amerikanern dazu, ein Amiga-Spiel zu unterbrechen, wenn der Chef den Arbeitsplatz umkreist, immer auf der Lauer nach Mitarbeitern, die nicht mitziehen. Diese „Boss-Taste" unterbricht das Spiel und bringt ein *Spreadsheet* — zu deutsch: Kalkulationsblatt — mit allerlei Zahlen, Formeln und Berechnungen auf den Schirm. Das Spreadsheet ist nämlich *die* typische und traditionelle Geschäftsanwendung für PC's. Mit einem Spreadsheet auf dem Amiga-Monitor erwischt zu werden, zeugt von Professionalität, Geschäftssinn und Management-Fähigkeiten. Alle Amigos mit Ambitionen auf höhere Manage-

 © 1990 technicSupport

ment-Weihen sollten sich die folgenden Erläuterungen daher keinesfalls entgehen lassen!

Spreadsheets — gestern, heute und morgen

Einige Jahre bevor der Amiga das Licht der Welt erblickte, begann der Siegeszug des *Personal Computing*: Eigene, höchstpersönliche Rechenleistung auf dem Schreibtisch verfügbar zu haben, revolutionierte die Arbeitswelt. Eine Software namens `VisiCalc` trug erheblich dazu bei; sie war das erste erfolgreiche Spreadsheet-Programm. Wie alle Geniestreiche war auch dieser recht simpel: eine Matrix aus Spalten und Zeilen, in die Zahlenwerte, Formeln oder auch Texte eingegeben werden konnten und die als Basis für weitere Kalkulationen diente. Um eine individuell ausgerichtete Kostenkalkulation komplizierter Natur aufzustellen, war es fortan nicht mehr nötig, eigens ein Programm in einer höheren Programmiersprache zu erstellen. `VisiCalc` war einerseits so anpassungsfähig, andererseits so leistungsstark, daß Kalkulationen schnell und bequem einzugeben und zu bearbeiten waren. Änderte sich ein Wert, war es nicht mehr nötig, die ganze Kalkulation neu vorzunehmen. Man änderte einfach diesen Wert auf dem Spreadsheet, und der Rest wurde vom Programm besorgt. Auf diese Art war es auch problemlos möglich, verschiedene Szenarios im Sinne einer Was-Wäre-Wenn-Analyse auszuprobieren, um zu sehen, wie sich Ergebnisse in Abhängigkeit ihrer Variablen verhalten.

Wie so oft im PC-Geschäft fand sich für eine blendende Idee ein Nachahmer, der sich dann zum Branchenführer mauserte: `Lotus Developments` brachte mit `1-2-3` *das* Spreadsheet auf den Markt, das die Standards setzte. Leistungsfähige Makro-Funktionen gaben dem Benutzer die Möglichkeit, über die bloße Kalkulationsfunktion hinaus Eingabemasken und Ausgabeprogramme zu definieren. `1-2-3` wurde so oft verkauft, daß es seitdem als Standard gilt.

Da die Spreadsheets immer öfter als Grundlage für Geschäftsberichte, Artikel und schriftliche Analysen ausgedruckt wurden, kam schnell der Ruf nach Textverarbeitungsfunktionen auf. Die Reaktion kam in Form sogenannter „integrierter“ Softwarepakete wie etwa

Symphony, Open Access oder Framework. Diese Pakete beinhalteten Spreadsheet-, Textverarbeitungs-, Datenbank- sowie Grafik-Funktionen. Wer wollte, konnte über Kommunikationsmodule seine Arbeit sogar per DFÜ verschicken.

Dann wurden die PC's grafikfähig: Mit dem MacIntosh von Apple kam die Mausunterstützung, und mit Microsofts Programm Excel wurden Spreadsheets bunter, hübscher und per Pull-Down-Menüs auch leichter bedienbar. Für die unmittelbare Zukunft sind noch weitere Verbesserungen angesagt: Lotus 1-2-3 in der Version 3.0 wird die dritte Dimension in Form von 3-D Spreadsheets bringen. Und für den wirklich professionellen PC-Anwender werden Spreadsheets in Zukunft ein Fenster zu SQL-Datenbanken (SQL = *Structured Query Language*, Datenbankabfragesprache) bringen, welche die Daten, z. B. von einem Großrechner, direkt in ein Spreadsheet übernimmt.

Woraus besteht nun das klassische Kalkulationsblatt? Ein Spreadsheet ähnelt einer Tabelle, die aus Zeilen und Spalten zusammengesetzt wird. Die Zeilen sind fortlaufend numeriert; die Spalten werden durch Buchstaben bezeichnet, wobei nach dem letzten Buchstaben Z eine zweite Stelle hinzugefügt wird, also es geht noch weiter mit AA, AB, AC usw. Die letzte unter VC verfügbare Spalte ist die Spalte AN und die letzte Zeile ist die Zeile 190. Die Kästchen, die jeweils einer Spalte und einer Zeile zugeordnet werden, nennt man im Spreadsheet-Jargon *Zellen*. Diese Zellen tragen als Adresse die Angaben über die zugehörigen Spalten und Zeilen, man spricht also von Zelle B18 oder Zelle F4. In diese Zellen werden Einträge als Zahlenwerte, Formeln oder Texte vorgenommen. Äußerlich ähneln sich die meisten Spreadsheets, so daß der Umstieg z. B. von Lotus 1-2-3 auf ein Amiga-PD-Spreadsheet nicht allzu schwer fallen dürfte.

VC — Spreadsheet-Programm für Einsteiger

Anhand des Programms VC soll in das Arbeiten mit Spreadsheets eingeführt werden. VC hat den Vorteil, einfach installierbar und ohne weitere Hardwareerweiterungen auf jedem Amiga lauffähig zu sein. Dafür ist es, um es gleich vorweg zu sagen, vom Bildschirmaufbau

 © 1990 technicSupport

her eher schlicht, und auch die Eingabe ist nicht immer die allerbequemste. Für Amiga-User interessant ist die Tatsache, daß VC in der vorliegenden Version die Intuition-Oberfläche, d. h. die Maussteuerung über Pull-Down-Menüs, unterstützt.

Der Aufruf von VC ist denkbar einfach. Auf der PD-Disk liegt bereits eine lauffähige Version vor, deren Icon einfach angeklickt werden muß. Das Programm öffnet nicht etwa nur ein Fenster, wie die meisten PD-Programme, sondern es initialisiert einen eigenen Screen in den aktuellen Workbench-Farben. Dies ist deshalb ein wichtiger Unterschied, weil sich ein Fenster beim Multitasking-Betrieb bekanntlich verkleinern und verschieben läßt. Den VC-Screen kann man aber leider nur nach unten wegziehen, was den Multitasking-Betrieb etwas erschwert, denn es wäre ja wünschenswert, die Werte aus dem Spreadsheet in eine Textverarbeitung zu übernehmen.

Bei Programmstart gliedert sich der VC-Screen in verschiedene Bereiche: Unterhalb der Titelzeile befindet sich die für den Benutzer wichtige Eingabezeile. Ist die aktuelle Zelle des Spreadsheets leer, stehen hier zwei eckige Klammern. Wenn die aktuelle Zelle (also die Zelle, auf der der Cursor gerade steht) beschrieben ist, erscheint in der Eingabezeile der Wert dieser Zelle. Dieser Wert kann bei Bedarf auch editiert, also verändert werden. In die leere Zelle können folgende Arten von Eingaben geschrieben werden: Soll ein mathematischer Wert, eine Zahl, eingegeben werden, ist zuerst das Gleichheitszeichen [=] einzugeben. Damit weiß das Programm, daß es sich bei der folgenden Eingabe um einen mathematischen Wert handelt. Danach wird der Wert eingegeben und mit [Return] abgeschlossen. Sogleich erscheint der eingegebene Wert in der entsprechenden Zelle des Spreadsheets. Bis zu acht Nachkommastellen können eingegeben werden; es ist also bei VC möglich, sehr genau zu arbeiten und zu kalkulieren. Bei der Notation ist zu beachten, daß die englische Schreibweise verwendet wird, also 3.567 und nicht 3,567. Texteingaben können linksbündig, rechtsbündig oder zentriert eingegeben werden. Soll der Text ganz links in der Zelle stehen, muß zuerst ein Kleiner-Zeichen [<] eingetippt werden, anschließend der Text, der im Spreadsheet erscheinen wird. Die Linksbündigkeit wird durch die Meldung Leftstring A1 angezeigt. Rechtsbündige Text-

eingaben werden mit [>] begonnen. Zentrierter Text wird mit dem Anführungszeichen initialisiert. Dann erscheint die Meldung `Label A1`.

Durch Eingabe von [X] kann der Inhalt der aktuellen Zelle gelöscht werden, egal ob darin Zahlenwerte, Formeln oder Texte stehen. (Hinweis: [X] meint hier die Betätigung der entsprechend beschrifteten Taste, also die Eingabe eines kleinen `x`.) Auch nachträgliches Editieren ist möglich, wenn auch nur in sehr eingeschränkter Weise: Durch Eingabe von [E] kann der Zelleninhalt verändert werden, allerdings nur mit der Backspace-Taste. Schön wäre es gewesen, wenn man sich mit den Pfeiltasten auf der Zeile hätte bewegen können, ohne die Zeichen zu löschen.

Der Inhalt einer Zelle kann in eine andere kopiert werden, indem die Ausgangszelle vom Cursor angefahren und dann durch Betätigen der [M] -Taste markiert wird. Danach wird mit dem Cursor die Zielzelle angefahren. Nach Betätigung der [C] -Taste erscheint in der Zielzelle der entsprechende Wert. Dies ist zwar eine recht bequeme und vor allem schnelle Methode des Kopierens, besonders wenn man es beispielsweise mit `Lotus 1-2-3` vergleicht, doch wird die markierte Zelle leider in keiner Weise angezeigt, so daß man nicht weiß, ob eine Zelle nun markiert ist oder nicht. Soweit zur Eingabezeile, die gleichzeitig als Anzeigeinstrument für den Inhalt der aktuellen Zelle fungiert. Soll diese Anzeige unterbleiben, kann man die Tastenkombination [CTRL] - [T] als Ein-/Aus-Schalter für diese Funktion verwenden.

Nachdem wir uns mit den verschiedenen Formen der Eingabe beschäftigt haben, soll das eigentliche Arbeiten mit dem Spreadsheet vorgestellt werden. Ein erstes, simples Kalkulationsbeispiel: die Addition einer Einkaufsliste. Fahren Sie den Cursor unter Benutzung der Pfeiltasten an eine beliebige Stelle des Spreadsheets, sagen wir einmal `C4`. Dort tragen wir den ersten Wert ein: `7.14`. Also zunächst, wie oben beschrieben, das Gleichheitszeichen drücken, damit `VC` weiß, daß nun die Eingabe eines numerischen Wertes erfolgt. Dann den Wert in der Notation `7.14` eingeben und [Return] drücken. Unser Wert erscheint in der Zelle `C4`. Zur Eingabe des nächsten Wertes den Cursor auf `C5` fahren, das Gleichheitszeichen drücken und einen

 © 1990 technicSupport

Wert, z. B. `14.35`, eingeben. In `C6` geben wir `123.13` ein, in `C7` `2.12` und in `C8` `34.16`, das reicht fürs erste. Die Werte stehen schön sauber untereinander. Nun sollen sie addiert werden. Ziehen wir in `C9` zunächst einen dicken Strich unter die Werte, damit unsere Addition auch optisch einen guten Eindruck hinterläßt. Dazu den Cursor nach `C9` fahren, das Anführungszeichen eingeben und neunmal das Gleichheitszeichen als Text eingeben. In `C10` soll die Addition erfolgen; wir geben dazu eine Formel ein. Formeln werden grundsätzlich mit dem *at*-Zeichen `@`, auch Klammeraffe genannt, eingeleitet. Also zunächst ein Gleichheitszeichen eingeben, dann das `@`-Zeichen und dann, ohne Leerzeichen, die Buchstaben `sum`. Als Argument geben wir dann einen Bereich des Spreadsheets an, der addiert werden soll. Wir wollen alle Zahlen von `C4` bis einschließlich `C8` addieren, also geben wir ein: `@sum(c4:c8)`. Und es ist jetzt wie Magie: In der Zelle `C10` steht die Summe, die wir errechnen wollten!

Stellen wir uns nun einmal vor, die oben errechnete Summe sei die Kostenaufstellung für eine Fete, die von drei Personen gleichmäßig getragen werden soll. Dazu müssen wir diese Summe durch drei Teilen. Wir fahren den Cursor auf das Feld `C10`, wo unsere Summenformel bekanntlich steht. Wenn wir nun ein kleines [E] eingeben, erscheint unsere Summenformel `@sum(c4:c8)`. An diese Formel hängen wir einfach unsere Rechenoperation dran, also `@sum(c4:c8)/3`. Der Slash „/“ dient also als Divisionszeichen. Wenn wir nun diese Eingabe mit [Return] abschließen, erscheint der neue, durch die Zahl 3 geteilte Wert in der Zelle `C10`. Natürlich können wir auch multiplizieren: Gesetzt den Fall, daß eine solche Fete sechsmal im Jahr stattfinden soll, können wir den jährlichen Aufwand pro Kopf berechnen lassen, also `@sum(c4:c8)*6/3`. Folgende Rechenoperationen sind also mit `VC` möglich:

`+` Addition

`-` Subtraktion

`*` Multiplikation

`/` Division

Als `@`-Funktionen steht neben der oben gezeigten Summenfunktion `@sum(z:z)` auch eine Produktfunktion `@prod(z:z)` zur Verfügung,

```
DBW_VC V1.0 (v1.1 880724 Amigaized by William E. Henning)
[(@sum(C4:C8))]

        A        B        C        D        E        F        G
0
1
2
3
4       Selters          7.14 DM
5       O-Saft          14.35 DM
6       Drinks         123.13 DM
7       Chips            2.12 DM
8       Cola            34.16 DM
9                  =============
10      Gesamt         180.90 DM
11
12
13
14
15
16
17
18
19
```

Mit VC lassen sich einfache Kalkulationen bewältigen

welche alle angegebenen Zellen multipliziert. Eine Durchschnittsfunktion `@avg(z:z)` ist ebenfalls dokumentiert, doch gelang es mir nicht, diese Funktion aufzurufen.

Nehmen wir einmal an, daß einer der Summanden aus dem Zellenbereich `C4` bis `C8` sich ändert, z. B. weil wir einen falschen Wert eingebgeben haben. Das Schöne beim Spreadsheet ist, daß wir nur diesen Wert zu ändern brauchen, und das Ergebnis wird automatisch neu berechnet. Ändern wir z. B. den Wert in `C5` , indem wir mit dem Cursor auf `C5` fahren, das kleine [E] eingeben und mit Backspace den alten Wert von `14.35` löschen. Wenn wir nun stattdessen `143.50` eingeben, ändert sich der Wert in `C10` automatisch. Das heißt: Wenn einmal die Struktur eines Spreadsheets festgelegt ist, lassen sich die Werte flexibel und dynamisch an neue Situationen anpassen. Angenommen, wir errechnen per Spreadsheet die monatlichen privaten Fixkosten für Auto, Miete usw. Wenn sich nun die Miete erhöht, muß nicht das ganze Spreadsheet neu definiert, sondern nur dieser eine Wert verändert werden, und wir sind

 © 1990 technicSupport

wieder auf dem neuesten Stand. Auf diese Weise läßt sich prima planen, und zwar im Sinne einer einfachen Was-Wäre-Wenn-Analyse: Wenn ich, um beim obigen Beispiel zu bleiben, unter dem Strich 200 DM für die Fete ausgeben will, wieviel darf ich dann für Bier, wieviel für Cola und wieviel für Würstchen ausgeben? Durch einfaches Ausprobieren verschiedener Werte im Spreadsheet lassen sich beliebige Möglichkeiten durchspielen, bis man schließlich die optimale Kostenverteilung ermittelt hat.

Auch die Struktur des Spreadsheets selbst läßt sich noch nachträglich ändern, gesetzt den Fall, es kommen noch weitere Positionen zu unserer Addition hinzu. Da wir unsere Zahlenreihe erst bei `C4` anfingen, könnten wir theoretisch noch in `C1` bis `C3` Werte eintragen und dann die Formel in `C10` auf `@sum(c1:c8)` ändern. Doch was, wenn mehr als drei neue Werte nachzutragen sind? Es gibt einen eleganten Weg, dies zu realisieren: Durch den Tastaturbefehl [I] (*insert — einfügen*) können an der aktuellen Cursorposition entweder Zeilen oder Spalten eingefügt werden. Geben Sie `ir` (*Insert Row*) ein, und es wird eine neue, leere Zeile eingefügt; durch `ic` (*Insert Column*) erzeugen Sie eine neue Spalte. Und nun der Clou: Das Spreadsheet ist intelligent und denkt mit! Wenn Sie eine neue Zeile eingefügt haben, ändert das Spreadsheet den Bereich der Summanden automatisch. Die Formel in `C10` wird, ohne daß der Benutzer es merkt, um eine Zeile erweitert: auf `@sum(c4:c9)`. Umgekehrt gibt es auch Befehle zum Löschen von Zeilen oder Spalten. Mit `dr` (*Delete Row*) kann der Benutzer eine Zeile löschen, mit `dc` (*Delete Column*) eine Spalte. Versehentliches Löschen kann rückgängig gemacht werden: Mit `pr` (*Pull Row*) wird die zuletzt gelöschte Zeile zurückgeholt, mit `pc` (*Pull Column*) die zuletzt gelöschte Spalte. Dabei wird automatisch die entsprechende Zahl von Spalten und Zeilen eingefügt, damit es nicht zum Überschreiben existierender Datenbestände kommt. Soll ein Zellenbereich dennoch überschrieben werden, steht der Befehl `pm` (*Pull Merge*) zur Verfügung, der die zuletzt gelöschte Zeile oder Spalte wieder ins Spreadsheet zurückholt, ohne Rücksicht auf Verluste durch Überschreiben.

Haben Sie Angst, daß Ihnen jemand bei der Arbeit über die Schulter schaut und geheime Daten wie Einkommen, Vorstrafen o. ä. zu

sehen bekommt, dann gibt es Befehle, die einen Zellenbereich unsichtbar machen. Trotzdem können diese Zellen in die Kalkulation einbezogen werden. `zr` (*Zap Row*) versteckt eine Zeile und `zc` (*Zap Column*) eine Spalte. Dieser Vorgang wird mit `sr` (*Show Row*) bzw. `sc` (*Show Column*) wieder rückgängig gemacht. Die Befehle `vr` und `vc` löschen in den Zellen befindliche Ausdrücke und Formeln, so daß nur noch der errechnete Wert darin steht. Wenn Sie also am Ende einer Kalkulation nur am Ergebnis und nicht am Rechenweg interessiert sind, können Sie den Zelleninhalt auf diese Weise konvertieren.

Der [F] -Befehl ist ein sehr nützlicher Befehl. Man benutzt ihn, um die Breite einer Spalte zu bestimmen. Die Spalten sind auf 10 Zeichen Breite voreingestellt. Beim Rechnen mit großen Werten oder bei der Texteingabe ist es aber oft nötig, breitere Spalten zu definieren, andernorts reichen auch schmalere Spalten. Wenn Sie [F] eingeben, werden Sie aufgefordert, die Breite neu zu bestimmen und zusätzlich die Zahl der Nachkommastellen (Voreinstellung: 8) festzulegen. Beim Rechnen mit Mark und Pfennig z. B. reichen zwei Nachkommastellen ja völlig aus.

Das Programm `VC` kann auf verschiedene Weise verlassen werden. Zum einen mit [CTRL]-[C] und zum anderen durch Eingabe von [Q] . Das Programm fragt gegebenenfalls ab, ob Änderungen am Spreadsheet abgespeichert werden sollen, und schließt den Screen ordnungsgemäß ohne Intervention irgendwelcher Gurus. Mit dem Fragezeichen kann ein bildschirmgroßes Hilfemenü aufgerufen werden, das alle Befehle und Funktionen zusammenfaßt.

Die Menüführung

`VC`-Versionen, die sich auf die oben beschriebene Art und Weise bedienen lassen, gibt es schon seit längerem auf diversen PD-Disks. Die hier vorgestellte Version glänzt aber zusätzlich durch Ausnutzung der Amiga-typischen Intuition-Bedienerstruktur. Beim Betätigen der rechten Maustaste erscheint in der Kopfzeile des Bildschirms die Menüzeile. Es stehen sechs Pull-Down-Menüs zur Verfügung:

Im `Project`-Menü steht zunächst einmal der Befehl `About`, der Informationen über Programmversion und Programmierer bereitstellt.

 © 1990 technicSupport

Die Befehle **Open** und **Save** laden bereits bearbeitete Spreadsheets bzw. speichern diese ab. Dabei kann auch ein Pfadname auf externe Massenspeicher angegeben werden, also z. B. **df1:spreadsheet** oder **df0:spreadsheets/kalkulation1**. Rechts neben den Befehlen im Pull-Down-Menü stehen Tastaturkürzel für den geübteren Benutzer. Wer es sich also merken kann und möchte, daß mit [Shift]-[G] ein Spreadsheet geladen wird, kann fortan auf den Aufruf mit der Maus verzichten. Der **Merge**-Befehl ([Shift]-[M]) fügt ein zweites Spreadsheet in das aktuelle ein. So ist es möglich, zunächst Teilbereiche einer Aufgabenstellung zu bearbeiten und alle diese Teile anschließend zu einem großen Spreadsheet zusammenzufassen. Zum Ausdrucken steht der Print-Befehl mit dem Tastaturkürzel [Shift]-[W] zur Verfügung. Das Programm fragt ab, welcher Bereich des Spreadsheets zum Drucker geschickt wird. Dazu ist zunächst die linke obere Ecke des Zellenbereichs anzugeben, anschließend die rechte untere Zelladresse. Die Druckausgabe erfolgt über den Drukker, der in den **Preferences** festgelegt wurde. Der Befehl **Table** oder kurz [Shift]-[T] druckt das Spreadsheet in Tabellenform mit einem Doppelpunkt als Trennsymbol zwischen den Zellen. Der letzte Befehl im Project-Menü, **Quit** dient zum Verlassen des Programms.

Das **Edit**-Menü hat zwei Befehle, jeweils mit Tastaturalternativen. Der **Copy**-Befehl ([/]) markiert einen Zelleninhalt, der **Paste**-Befehl (**pm**) schreibt diesen kopierten Zelleninhalt in eine andere Zelle.

In dem Menü, das mit **Cell** überschrieben ist, finden wir einige Befehle, die bereits oben als Tastaturbefehle vorgestellt wurden und hier deshalb nicht ausführlich behandelt werden müssen. **Label**, **Value**, **Clear**, **Copy** und **Mark** kennen wir schon. **Stat** bzw. [CTRL]-[T] zeigt den Status der Zelle in der obersten Zeile an. Mit dem Befehl **Stat** wird diese Anzeige ein- bzw. ausgeschaltet. **Type**, bzw. [CTRL]-[A], schreibt den Wert einer Zelle in die Kommandozeile. Die nächsten beiden Menüs, **Row** und **Col** enthalten die ebenfalls bereits bekannten Befehle zum Einfügen, Formatieren usw., so daß hier nicht weiter darauf eingegangen werden muß.

Es ist müßig, über Sinn und Unsinn des sechsten Menüs mit dem Namen `Cursor` zu spekulieren. Es enthält Befehle zur Cursorsteuerung, d. h. sie steuern das, was eigentlich viel einfacher mit den Pfeiltasten zu regeln ist. Allenfalls die Befehle `First`, mit dem der Cursor auf Spalte 0 gesetzt wird, `Last`, womit der Cursor auf die letzte Spalte gesetzt wird, sowie `Redraw`, mit dem das Spreadsheet neu berechnet wird, haben einen Sinn.

Soweit zu den Funktionen und Möglichkeiten, die das Public Domain Spreadsheet `VC` bietet. Sicherlich wird es einige Zeit benötigen, bis ein ungeübter Anwender alle Feinheiten des Arbeiten mit Spreadsheets kennengelernt hat. Doch gerade für einen Benutzer, der die Investition in ein professionelles Spreadsheet-Programm, von denen es auf dem kommerziellen Sektor ja einige gibt, scheut, wird in `VC` ein kostengünstiges Programm finden, welches zwar keine professionelle Qualität erreicht, zum Lernen und Einarbeiten in die Materie jedoch gut geeignet ist.

AnalytiCalc — Spreadsheets für Profis

Von den beiden hier vorgestellten PD-Spreadsheets ist `AnalytiCalc` das mit dem höheren Anspruch an Professionalität. Dies hat seine negativen Seiten darin, daß es sehr umfangreich, speicherintensiv und zudem nicht leicht zu installieren ist. Auch ist es empfohlen, den Interlace-Modus vor dem Programmstart zu aktivieren, um alle Feinheiten der Bildschirmdarstellung ausnutzen zu können. Die Installation gestaltet sich deshalb schwierig, weil auf der Fishdisk 176 nur Programmdateien zu finden sind, die komprimiert wurden, um alles auf einer Diskette unterbringen zu können. Diese Komprimierung muß zuerst rückgängig gemacht werden, was zumindest ein weiteres Diskettenlaufwerk, oder 1MB Ram-Speicher; idealerweise aber eine Festplatte erforderlich macht.

Installation auf Festplatte

Mit einem PD-Programm namens `BOOZ` müssen die Dateien der Fishdisk mit dem Anhang `.ZOO` zunächst dekomprimiert werden. Auf der Platte wird ein Unterverzeichnis eingerichtet, z. B.

 © 1990 technicSupport

`dh0:AnalytiCalc`. Dorthinein werden alle `.ZOO`-Dateien kopiert, wie auch das `BOOZ`-Programm selbst, das sich im `C`-Verzeichnis der Fishdisk befindet. Dann wird unser Verzeichnis mit

```
cd dh0:AnalytiCalc
```

als aktuelles Verzeichnis ausgewählt. Darauf wird dekomprimiert. Die Syntax lautet:

`BOOZ X` ⟨*Datei*⟩`.ZOO`

also beispielsweise

```
BOOZ X AnalytiCalc.ZOO
```

Das BOOZ-Programm stellt dann automatisch die Dateien in ihrer ursprünglichen Form, also auch mit ihren Originalnamen, her. In gleicher Weise wird mit den anderen vier `.ZOO`-Dateien verfahren. Anschließend können die `.ZOO`-Dateien sowie das BOOZ-Programm gelöscht werden. Was übrigbleibt, ist eine lauffähige Version von `AnalytiCalc`, die mit `Analy` aufgerufen wird.

Installation mit Hilfe der Ram-Disk

Wer keine Festplatte besitzt, aber sehr wohl über 1MByte oder mehr verfügt, kann sich mit Hilfe einer RAM-Disk eine Diskette mit einer lauffähigen Version des Programms erstellen. Dazu wird mit

```
copy BOOZ ram:
```

das `BOOZ`-Utility kopiert. Aus Platzgründen sollte anschließend eine `.ZOO`-Datei nach der anderen auf der RAM-Disk dekomprimiert werden. Dazu zunächst mit

`copy` ⟨*Datei*⟩`.ZOO ram:`

die erste Datei ins Ram kopieren, anschließend mit

`cd ram:`
`BOOZ X` ⟨*datei*⟩`.ZOO`

die Ursprungsdatei wiederherstellen. Dann wird mit

`delete` ⟨*Datei*⟩`.ZOO`

die erste .ZOO-Datei gelöscht. Die Programmdateien sollten erst auf eine formatierte Diskette abgespeichert und dann in der Ram-Disk gelöscht werden. Textdateien und Manuals müssen auf eine separate Diskette kopiert werden, um den kostbaren Platz auf der Programmdiskette zu schonen. In gleicher Weise kann nun mit den übrigen vier .ZOO-Dateien verfahren werden.

Wenn man sich mit 512 KByte beschränken muß, reicht der maximale Speicherplatz der Ram-Disk nicht immer zum Dekomprimieren der Dateien aus. In diesem Fall kann man den Vorgang des Dekomprimierens statt in der Ram-Disk auf einer formatierten Leerdiskette ausführen.

Aufruf des Programms

Bei regelmäßigem Gebrauch von **AnalytiCalc** empfiehlt es sich, eine speziell auf dieses Programm zugeschnittene Workbench-Diskette anzufertigen. Dazu fertigt man eine Kopie der normalen Workbench an und stellt auf dieser Kopie die **Preferences** so ein, daß nach dem Booten der Interlace-Modus aktiv ist. Im Preferences-Menü also das Interlace-Gadget auf [ON] stellen und mit [SAVE] abspeichern. Danach sollte die **Startup-Sequence** editiert werden:

1. Alle Treiber, Utilities und residente Programme, die nicht unbedingt nötig sind, aus der **Startup-Sequence** entfernen, falls nicht mehr als 1MByte vorhanden sind.
2. Den Befehl **Stack 25000** einbinden. Mit 512KB kann es bei dieser Stack-Größe Probleme geben; gegebenenfalls muß der Wert herabgesetzt werden.
3. Um weiteren Speicherplatz zu sparen, kann der Befehl **LoadWB** herausgenommen werden.
4. An Stelle dieses Befehls kann der Programmaufruf direkt in die **Startup-Sequence** eingebunden werden, also **dh0:[Verzeichnis/]analy** beim Starten von der Festplatte bzw. **df1:analy** vom Zweitlaufwerk.

Mit dieser Art der Installation ist das Programm direkt nach dem Booten aktiv. Das Programm fragt zuerst ab, ob die Interlace-

 © 1990 technicSupport

Auflösung von 640 × 400 Punkten besteht. Wird ein [N] für „no“ eingegeben, nimmt das Spreadsheet nur 600 × 200 Punkte ein. Im oberen der beiden Fenster, die dann erscheinen, befindet sich unser Spreadsheet.

Vorsicht! Manche Textdateien auf der Fishdisk schlagen einen anderen Installationsweg unter Benutzung anderer Dekomprimierungsmechanismen vor. Dies wird nicht möglich sein, da auf der Fishdisk die dazu nötigen Dateien nicht vorhanden sind.

Erste Schritte mit AnalytiCalc

Wer denkt, daß nach der komplizierten Installationsarbeit nun das bequeme Arbeiten mit dem Programm beginnen kann, hat sich geirrt. `AnalytiCalc` ist ein so mächtiges und umfangreiches Instrument, daß dies auch auf Kosten der Bedienbarkeit geht. Man muß dies aus der Geschichte des Programms verstehen: `AnalytiCalc` wurde nicht im Hinblick auf die spezifischen Amiga-Möglichkeiten konzipiert, sondern erst im nachhinein angepaßt. Es wurde in Fortran programmiert, einer mathematisch orientierten Großrechnersprache. Eine komfortable Intuition-Steuerung mit Mausunterstützung, Pull-Down-Menüs usw. kann man also vergessen. Alle Befehle sind tastatur- und zeilenorientiert.

Andererseits gehört `AnalytiCalc` mit seinem riesigen Befehls- und Funktionsumfang zu den umfangreichsten Spreadsheets überhaupt. Alle Möglichkeiten aufzuzählen, würde den Rahmen dieses Buches sprengen, was allein die Tatsache unterstreicht, daß dem Programmpaket eine Dokumentationsdatei von 105(!) Seiten Umfang beiliegt. Wir können an dieser Stelle also nur einige Prinzipien der Benutzerführung aufzeigen. Nach dem Programmaufruf stehen zwei Fenster zur Verfügung, wobei das obere, größere Fenster unser Spreadsheet enthält. Wie bereits gesagt, können wir die Maus jetzt beiseite legen, denn alle Benutzerfunktionen sind an die Steuerung eines VT100-Terminals angelehnt. Das Programm kennt zwei Befehlsmodi: Im ersten Modus erwartet das Programm zunächst einen Befehl und dann Argumente. Eine Eingabe kann also nur erfolgen, indem der Befehl für „Eingabe“, gefolgt von dem Eingabewert selbst,

eingetippt wird. `E 213` würde z. B. den Wert 213 in die aktuelle Zelle schreiben. Alternativ dazu kann ein Modus gewählt werden, der Eingaben direkt als Zelleingaben interpretiert. Dieser Modus wird durch Eingabe eines Slash (/) aktiviert. Der Prompt ändert sich von „>“ in „:“. Unsere 213 könnte dann direkt eingetippt und mit Return abgeschickt werden. Mit zwei Slashs (//) wechseln wir zurück in den ersten Modus.

Der professionelle Anspruch des Programms wird durch die maximale Spreadsheetgröße unterstrichen: Ein Zeilenbereich von 1–18000 und ein Spaltenbereich von A–ZPH sind adressierbar. Das sollte selbst für allerhöchste Ansprüche ausreichen. Darüber hinaus ist es möglich, von einer Zelle aus eine Zelle eines anderen Spreadsheets anzusprechen, das auf Diskette vorliegt. Auf diese Weise lassen sich sogar mehrere Spreadsheets verknüpfen.

AnalytiCalc umfaßt 85 Funktionen und 70 Kommandos, deren Variationen sich hier nicht aufzählen lassen. Daher befassen wir uns kurz mit den **Insert**-Variationen, um die Arbeitsweise exemplarisch darzustellen. Wer sich wirklich die professionellen Möglichkeiten des Programms zunutze machen möchte, kann sich über [Alt]-[F1] eine Online-Aufzählung der wichtigsten Befehle aufrufen oder durch Eingabe von **Help** eine umfassendere Hilfsfunktion nutzen. Im Falle von **VC** haben wir gesehen, daß es oft nötig ist, Zeilen oder Spalten in ein bereits existierendes Spreadsheet einzufügen. Bei **AnalytiCalc** geschieht dies durch den **Add**-Befehl. Er muß, wie bereits angedeutet, als Tastaturbefehl eingegeben werden und kann folgende Varianten annehmen:

AA ⟨*Zahl*⟩ **R**
AA ⟨*Zahl*⟩ **C**
AR ⟨*Zahl*⟩ **R**
AR ⟨*Zahl*⟩

Die erste Variante fügt über der aktuellen Cursorposition die Anzahl von Zeilen ein, die durch das Argument ⟨*Zahl*⟩ bestimmt wurde, also 9 Zeilen bei **AA 9 R**. Die zweite tut das gleiche für die Anzahl von Spalten, die links vom Cursor eingefügt werden sollen. Hierbei wird der Rest des alten Spreadsheets nach rechts, bzw.

 © 1990 technicSupport

nach unten verschoben, und zwar so, daß eventuelle Zellreferenzen nicht geändert werden. Das muß genau beachtet werden, denn wenn im verschobenen Bereich auf eine oder mehrere Zellen verwiesen wird (z. B. wenn eine Summe von Zellen berechnet werden soll), können sich Fehler ergeben. Die dritte und vierte Variante aktualisieren deshalb automatisch diese Zellreferenzen, um solche Fehler auszuschließen. Wie ist diese Art der Steuerung zu bewerten? Das Positive daran ist, daß ein großer Vorrat an mächtigen Befehlen zur Verfügung steht. Andererseits erfordert dies eine genaue und langwierige Einarbeitung. Der Benutzer kann sich nicht einfach vor den Computer setzen und erwarten, daß er das Programm auf Anhieb beherrscht. Ein sorgfältiges Erlernen der Befehle ist nötig, und es wird etwas dauern, bis das Programm in aller Fülle benutzbar ist. Vielleicht wäre es günstig, `AnalytiCalc` durch eine Intuition-Oberfläche leichter bedienbar zu machen. Problematisch dabei ist die Tatsache, daß die Funktion des Befehls-Journals nicht mehr einsetzbar wäre. Diese Funktionen erlaubt es dem Benutzer, Befehls-Makros zu generieren, indem Tastatureingaben aufgezeichnet und abgespeichert werden und dieselbe Befehlsfolge somit mehrmals aufrufbar ist. Das wäre bei Menüaufrufen so nicht möglich. Selbst aus diesen kurzen Einblicken in die Funktionsweise des Programms dürfte der enorme Funktionsbestand ersichtlich sein. Deshalb ist auch nur dem Spreadsheet-erfahrenen Benutzer anzuraten, sich mit `AnalytiCalc` auseinanderzusetzen. Wer sich allerdings die Mühe gemacht hat, sich richtig einzuarbeiten und das 105-seitige Handbuch auszudrucken und zu studieren, wird ein Spreadsheet für absolute Power-User vorfinden, das hinsichtlich der Funktionalität an professionelle Programme heranreicht — bis auf die fehlende Intuition-Unterstützung.

HERKUNFT

Die hier vorgestellte Version 1.0 von `VC` ist auf der Panorama-Disk 32c zu finden. Die Programmierung von `VC` ist durch verschiedene Hände gegangen. Die erste Version von JAMES GOSLING ist auf Fishdisk 53 zu finden und wurde von MARK WEISER modifiziert. Ein gewisser BOB BOND programmierte weitere Zusätze, bis sie schließlich DAVE WECKER in die Hände bekam. BILL HENNING

stellte die nun vorliegende Version auf seinem Bulletin Board System zur Verfügung, von wo sie in die Public Domain gelangte.

`AnalytiCalc` finden Sie in der komprimierten Form auf Fishdisk 176. Der Autor ist GLENN EVERHART. Er bittet alle Benutzer seines Programms um eine Gebühr von $10 an folgende Adresse:

```
Glenn C. Everhart
25 Sleigh Ride Rd.
Glen Mills
Pennsylvania 19342
USA
```

VERWEISE

Weitere PD-Tabellenkalkulationen sind `Spread` (TBAG 26), `SpreadSheet` (Fishdisk 53) und `VisiCalc` (RPD 90). Die Fähigkeit, Ausschnitte eines Spreadsheets direkt in Präsentationsgrafiken umzusetzen, bieten nur kommerzielle Programme wie beispielsweise `MaxiPlan Plus`, in einer Demo-Version zu begutachten auf der Faug-Disk 44.

© 1990 technicSupport

Vdraw

Malprogramm für mehrere Amiga-Grafikauflösungen

AUFRUF

Das Programm wird von der Workbench über das `VDraw`-Icon gestartet. Zu beachten ist, daß `VDraw` die Mathemathik-Bibliotheken der Workbench sowie einige eigene Dateien aus dem `VDraw`-Verzeichnis benötigt. Die Bibliotheken sollten sich im Verzeichnis `LIBS:`, alle anderen Zusatzdateien in dem mit dem `assign`-Befehl definierten logischen Gerät `VDraw:` befinden. Im Falle der Panorama 19a wäre aus dem CLI heraus

```
assign vdraw: df<x>:vdraw1.21
```

einzugeben. Ein anderer Weg bestünde darin, alle Dateien aus dem `VDraw`-Verzeichnis in das Hauptverzeichnis einer anderen Diskette zu kopieren und dieser den Namen „VDraw“ zu geben.

ZWECK

`VDraw` ist das ausgereifteste Malprogramm aus der Public-Domain-Fundgrube. Durch die Unterstützung vieler Auflösungen hebt es sich deutlich von anderen Malprogrammen ab. Leider wird der PAL-Modus in der Anzeige nicht unterstützt. Trotzdem lassen sich solche Bilder durch die Verwendung von Superbitmaps erstellen und abspeichern. Vor allem die Möglichkeit, sich seine eigene Benutzeroberfläche zu erstellen, und die vielen Utilities ließen `VDraw` zum Spitzenreiter unter den PD-Malprogrammen werden.

BESCHREIBUNG

`VDraw` lädt beim Programmstart mehrere Dateien nach, deren Funktion im folgenden erläutert werden soll:

„LIBS:mathtrans.library“ sowie „LIBS:mathffp.library“ dienen zur Berechnung einiger Spezialeffekte und befinden sich normalerweise im `Libs:`-Verzeichnis der Workbench. „VDraw:VDraw.extra_icons“, „VDraw:VDraw.arrows_icons“ und „VDraw:VDraw.tools_icons“

stellen einfarbige IFF-Bilder vordefinierter Größe dar. Ausschnitte dieser Bilder fungieren im Programm als Gadgets, so daß sich der Benutzer durch Änderung dieser Bilder eine eigene Oberfläche schaffen kann. „VDraw:VDraw.arrows_icons“ wird zur Zeit aber noch nicht verwendet. Mit der Datei „VDraw:VDraw.profile“ hat es etwas ganz besonderes auf sich. Sie bestimmt die wichtigsten Parameter beim Start des Programmes und hat folgendes Format:

```
Verhältnisquotient (Aspect Ratio)
Breite der SuperBitmap//Höhe der SuperBitmap//Zahl der Bitplanes
5 unbenutze Felder (für spätere Pfeil-Kopf-Anwendungen)
Erneuerungsrate für Zoom-Modus in Mikrosekunden
Breite des Bildschirms // Höhe des Bildschirms
```

Ein Beispiel könnte also so aussehen:

```
600
960 720 1
7 -11
100000
640 400
```

Dies würde einen Bildschirm in NTSC-Auflösung und *Hires-Interlace* eröffnen. Die SuperBitmap (der Bereich, auf dem insgesamt gezeichnet werden kann) hätte genau die Auflösung eines Epson-Druckers — auf DIN A4 — gesehen und ist zweckmäßigerweise einfarbig (1 Bitplane). Der Verhältnisquotient erhält den Wert 600. Dadurch erscheinen Kreise auf dem Bildschirm auch auf dem Drucker als solche anstatt als Ovale. Die Berechnung erfolgt nach der Formel:

$$\frac{vertikale\ Auflösung\ in\ DPI}{horizontale\ Auflösung\ in\ DPI} * 1000$$

Bei einer Auflösung von $72 * 120DPI$ ergibt das den Wert $72/120 * 1000 = 600$. Der Standardwert sollte hier jedoch 1000 betragen und muß für jeden Drucker in jeder seiner Auflösungen neu berechnet werden. Als *Refresh*-Rate für den *Zoom*-Modus hat sich 100.000 als guter Arbeitswert erwiesen. Hier kann natürlich noch ein wenig experimentiert werden. Alle diese Einstellungen lassen sich nicht mehr ändern, nachdem `VDraw` geladen ist.

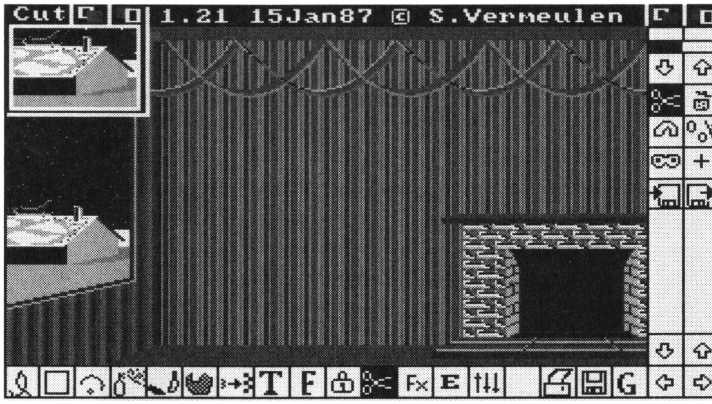
 © 1990 technicSupport

Bei Bedarf kann man sich mehrere „VDraw.profile"-Dateien erstellen, die dann unter verschiedenen Namen abgelegt und jeweils nur noch in „VDraw.profile" umbenannt werden müssen. So haben Sie einen schnellen Zugriff auf verschiedene Grundeinstellungen. Bedenken Sie jedoch, daß der CHIP-Speicher im Amiga durchaus begrenzt ist. Zu hohe Auflösungen oder riesige SuperBitmaps zollen Ihren Tribut an den Speicher und machen später vielleicht die Verwendung einiger Funktionen unmöglich — wenn sie nicht gar einen Absturz des Systems verursachen!

Auch die Anzahl der Bitplanes und damit die der Farben hat großen Einfluß auf den Speicherverbrauch zusammen. Die maximale Zahl darf unter HiRes 4 und unter LoRes 5 nicht überschreiten. Bekommt `VDraw` schon beim Start Probleme mit dem Speicher, ist das jedoch nicht weiter schlimm, da aller bis dahin allokierter Speicher wieder freigegeben und das Programm beendet wird. Daraufhin sollten Sie dann einige Tasks aus dem System entfernen, Laufwerke abklemmen oder in der „VDraw:VDraw.profile"-Datei die Auflösung und/oder die Anzahl der Farben verringern.

Doch kommen wir endlich zum eigentlichen Zweck des Programmes: dem Zeichnen. Gleich nach dem Start erscheinen ein neuer Bildschirm in der gewählten Auflösung und das Zeichenfenster. Am unteren Bildschirmrand (falls Sie einen PAL-Monitor besitzen ist es eher der mittlere) erscheint eine Reihe von Gadgets, die als Hauptmenüpunkte agieren und mehrere Unterpunkte fassen.

Diese Unterpunkte erscheinen am rechten Bildschirmrand, sobald Sie einen Hauptpunkt anwählen. Die Funktionen, die den einzelnen Menüs zugeordnet sind, werden in steter Reihenfolge — Hauptmenü von links nach rechts und Untermenü von oben nach unten, wobei erst linke, dann rechte Spalte — erläutert. Das macht diese Beschreibung gleichzeitig zu einem Nachschlagewerk, falls bei der Arbeit mit `VDraw` Fragen auftreten sollten. Bei der Zählung werden Pfeilgadgets im allgemeinen ausgeschlossen, da sie meist dieselbe Bedeutung haben.



Vdraw überragt alle anderen PD-Malprogramme

1. Freies Zeichnen
Dieser erste Menüpunkt stellt die Basisfunktionen des Malprogramms zur Verfügung.

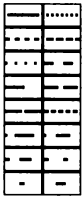
1.1 Freihandzeichnen
Hier folgt die Linie auf dem Bildschirm direkt den Bewegungen der Maus. Wenn die Bewegungen zu schnell werden und `VDraw` der Verlust von Punkten droht, werden die entsprechenden Punkte einfach durch gerade Linien ersetzt. Das sieht dann manchmal recht abgehackt aus, erlaubt aber auch interessante Effekte. Trotzdem ist es meist besser, wenn man langsamer zeichnet.

1.2 Linien zeichnen
Durch den ersten Mausklick wird der Startpunkt, durch den zweiten der Endpunkt der Linie markiert, die dann in der Farbe gezogen wird, die man in der linken oberen Bildschirmecke angewählt hat.

 © 1990 technicSupport

1.3 Vertikale und horizontale Linien
Ähnlich 1.2, mit dem Unterschied, daß automatisch nur senkrechte und waagerechte Linien gebildet werden.

1.4 Ohne Funktion
Eine noch nicht implementierte Funktion, die in Zusammenhang mit der „VDraw:VDraw.arrows_icons“ Datei stehen wird.



1.5 bis 1.20 Verschiedene Linienarten und -stärken
Die Effekte der verschiedenen Linientypen lassen sich am besten durch Ausprobieren ermitteln.

2. Rechtecke zeichnen
In diesem Menü findet man verschiedene Funktionen für das Zeichnen von Rechtecken.

2.1 Rechteckige Rahmen zeichnen
Der erste Mausklick im Zeichenfeld bezeichnet den einen, der zweite den anderen Eckpunkt des Rechteckes, das daraufhin mit Linien markiert wird.

2.2 Ausgefüllte Rechtecke zeichnen
Wie 2.1, nur werden die Rechtecke nun auch in der aktuellen Farbe ausgefüllt.

2.3 Mit Mustern ausgefüllte Rechtecke
Diese Funktion ist noch nicht implementiert.

2.4 Rechteck invertieren
Der markierte Bereich wird invertiert. Besonders mit mehreren Farben ergibt das oft überraschende Effekte.

3. Kreise zeichnen
Mit den Funktionen dieses Menüs lassen sich Kreise und vielfältige Formen von Kreisbögen erstellen.

3.1 Kreise
Der erste Mausklick bestimmt den Kreismittelpunkt. Nun kann die Maus bewegt werden, bis der Kreis die gewünschte Größe hat. Nach dem endgültigen Mausklick wird der Kreis dann „zu Bildschirm gebracht".

3.2 Kreisausschnitte
Hier werden Kreisausschnitte gezeichnet, deren drei Parameter (Anfang, Radius, Ende) durch gleichzeitiges Drükken der Maustaste und [Ctrl], linker [Shift] - oder linker [Alt] -Taste eingestellt werden. [Ctrl] verschiebt dabei den Anfang, die linke [Alt] -Taste das Ende des Kreisausschnittes. Die linke [Shift] -Taste vergrößert oder verkleinert den Radius — je nach Mausbewegung. Zweckmäßigerweise drückt man schon beim ersten Mausklick, der den Mittelpunkt angibt, die linke [Shift] -Taste und macht den Kreis durch Vergrößerung des Radius erst einmal sichtbar. Nun können Start- und Endpunkt eingestellt werden. Weitere Korrekturen der Parameter sind noch möglich, bis wieder ein Mausklick ausgeführt wird. Dann ist der Ausschnitt gezeichnet, und es können keine Änderungen mehr vorgenommen werden.

3.3 Bezier-Kurve mit einem Bezugspunkt
Der erste Mausklick bezeichnet den Anfang, der zweite das Ende der Kurve. Nun kann die Kurve durch Mausbewegungen auf bestimmte Arten verformt werden. Wenn das Ergebnis mit Ihren Wünschen übereinstimmt, bannt ein dritter Klick die Kurve in das Gemälde hinein.

3.4 Bezier-Kurve mit zwei Bezugspunkten
Auch hier wird die Länge der Kurve wieder mittels zwei Mausklicks festgelegt. Bei der nun folgenden Bewegung mit der Maus, sollte jedoch gleichzeitig die linke [Shift] - oder die linke [Alt] -Taste gedrückt werden, da diese Funktionen die Bezugspunkte für die Kurve verschieben. Nach einem dritten Mausklick ist auch diese Kurve fertig. Diese Zeichenart bietet mit Sicherheit nach kurzer Eingewöhnungszeit die flexibelste Lösung des Kurvenproblems.

© 1990 technicSupport

3.5 und 3.6 Ohne Funktion
Diese Funktionen sind in der Version 1.21 noch nicht implementiert.

4. Sprühdoseneffekte
Diese Funktion darf heute in keinem Zeichenprogramm mehr fehlen. Zufällig gesetzte Punkte, deren Dichte sich mit der Zeit auf einem Fleck erhöht, geben gerade bei `VDraw` einen erstaunlich echten Sprühdosen-Effekt. Statt eines Untermenüs erscheinen hier zwei Rollbalken. Mit `R` kann der Radius des Sprühkreises, mit `D` die Dichte beziehungsweise die Stärke der Farbe verändert werden.

5. Ausschnitte setzen
Ausschnitte (*Brushes*), die unter 11. gemacht oder eingeladen worden sind, können hier wieder ins Bild geklebt werden. Da `VDraw` mehrere Brushes im Speicher halten kann, ist der gewünschte Brush erst mittels der beiden Pfeilgadgets auszuwählen. Zu beachten ist, daß in diesem Modus das Verlassen von `VDraw` nicht funktioniert, da es ansonsten zu Speicherproblemen kommen kann. Um `VDraw` zu verlassen, müssen Sie sich in einem anderen Modus befinden.

5.1 Pinselfunktion
Der Ausschnitt wird als Pinsel behandelt und kann bei jedem Mausklick auch durchgezogene Linien hinterlassen.

5.2 Stempelfunktion
Der Pinsel wird als Stempel behandelt und kann pro Mausklick nur eine Kopie auf den Bildschirm „stanzen“.

6. Flächen füllen
Diese Funktion ist speicheraufwendig und kann unter Umständen bei Speichermangel aussetzen. Dies wird dann jedoch im `VDraw`-Fenster auf der Workbench dokumentiert. Also Achtung! `VDraw` kann Flächen nur mit definierten Mustern füllen. Diese Muster lassen sich als Brushes einladen

oder aus anderen Zeichnungen herausschneiden. Zweckmäßig ist ein Einladen der Datei „VDraw:VDrawbrush“, aus der dann beliebige Muster ausgeschnitten und verwendet werden können. Sind gerade mehrere Ausschnitte im Speicher vorhanden, muß der zu verwendende erst mittels der Pfeilgadgets ausgewählt werden. Ist das getan, ist die maximal zu füllende Fläche mittels einer Box zu markieren. Diese Angabe verhindert das Füllen des gesamten Bildschirmes, falls bei einem Füllvorgang mal etwas schiefgeht und in der zu füllenden Fläche ein Loch ist, das den Bildschirm zum „Überlaufen“ bringen könnte. Als letztes muß noch einmal in die eigentliche, zu füllende Fläche geklickt werden, um das Muster zu erhalten.

7. Lupenfunktion
Hier wird die linke, obere Ecke des Zeichenfensters 8-fach vergrößert. Punkte können mittels Mausklick gesetzt oder gelöscht werden. Der sichtbare Ausschnitt kann mit den vier Pfeilgadgets in der rechten unteren Ecke verschoben werden.

8. Texteingabe
In diesem Menüpunkt können sämtliche Amiga-Zeichensätze eingeladen und auf den Bildschirm gebracht werden. Der zu schreibende Text ist dazu in das Gadget am unteren Bildschirmrand einzutragen. Nachdem Sie [Return] gedrückt haben, kann der Text durch anhaltendes Drücken auf den linken Mausknopf über den Bildschirm bewegt werden. Erst wenn alle Attribute 8.1 bis 8.6 — auch Kombinationen sind möglich — eingestellt und [OK] angeklickt worden ist, werden alle Attribute sichtbar und der Text tatsächlich plaziert. Falls man es sich noch anders überlegt, kann aus diesem Menü immer noch mit [Quit] ausgestiegen werden.

8.1 Unterstreichen
Der Text wird unterstrichen dargestellt.

 © 1990 technicSupport

8.2 Fettschrift
Bei Anwahl dieses Gadgets wird der Text in Fettschrift geschrieben.

8.3 Italic
Dieses Gadget bewirkt eine Darstellung des Textes in Kursivschrift.

8.4 Extended
Falls der Zeichensatz ein solches Aussehen vorsieht, kann diese Sonderform hier angewählt werden.

8.5 Zeichen-Modus 1
Hier werden die Buchstaben auf das Bild aufgesetzt. Deshalb erscheinen um den Text herum noch einige Pixels in der Original-Hintergrundfarbe. Das erhöht in den meisten Fällen die Lesbarkeit des Textes.

8.6 Zeichen-Modus 2
In diesem Modus wird der entsprechende Text nur auf dem Bildschirm abgelegt. So kann es auch passieren, daß eine Linie mitten durch den Text läuft.

8.7 Komplementäre Schrift
Die Schriftfarben werden vertauscht.

8.8 Inverse Schrift
Die Textzeile wird in inverser Schrift ausgegeben.

9. Zeichensatz laden
Hier wird der Zeichensatz geladen, der unter 8 verwendet wird. Einfach den gewünschten Zeichensatz — er muß natürlich im `Fonts:`-Verzeichnis liegen — anklicken oder mit den Pfeilgadgets auswählen und mit 9.1 einladen.

9.1. Zeichensatz einladen
Der selektierte Zeichensatz wird geladen. Wenn es hierbei Speicherplatzprobleme gibt, wird das wie immer im Fenster auf der Workbench dokumentiert.

10. Teilbereiche sichern
Wichtige Teile der Zeichnung können zwischengespeichert werden, damit sie bei versehentlichen Veränderungen wieder hergestellt werden können. Dazu ist erst ein Rahmen in bekannter Art und Weise zu ziehen, der dann jedoch sofort wieder verschwindet. Lassen Sie sich dadurch nicht irritieren, und klicken Sie auf 10.1.

10.1 Sichern eines Bereiches
Der Rahmen erscheint wieder und zeigt dadurch an, daß dieser Ausschnitt gespeichert und angewählt ist. Auf dieselbe Weise lassen sich weitere Teilbereiche sichern. Falls es mehrere von ihnen gibt, muß der betreffende Ausschnitt, auf den sich die Funktion 10.2 oder 10.4 beziehen soll, immer mittels der Pfeilgadgets in den Rahmen gebracht werden.

10.2 Restaurieren eines gesicherten Ausschnittes
Der durch andere Funktionen veränderte, aber vorher mit 10.1 zwischengespeicherte Ausschnitt kann mit dieser Funktion wieder zurückgeholt werden. Dazu ist vorher der entsprechende Bildschirmausschnitt mit den Pfeilgadgets auszuwählen.

10.3 Ohne Funktion
Dieses Feature ist noch nicht implementiert.

10.4 Entsichern eines Ausschnittes
Soll ein zwischengespeicherter Bereich wieder freigegeben werden, wird er nach Anwahl mit den Pfeilgadgets und einem Klick auf dieses Gadget wieder normal behandelt.

11. Ausschnitte (Brushes) erstellen
Mit der Maus kann ein Teil des Bildes als Ausschnitt definiert und als Pinsel oder Füllmuster verwendet werden. Auch ein Abspeichern auf Diskette ist möglich. Auch hier führen Sie die Pfeil-Gadgets durch alle schon vorher definierten Ausschnitte hindurch.

© 1990 technicSupport

11.1 Pinsel definieren
Über ein Rechteck wird ein beliebig großer Ausschnitt als Pinsel definiert.

11.2 Pinsel löschen
Ein Pinsel wird wieder gelöscht. Zuvor muß der gewünschte Pinsel mit den Pfeil-Gadgets selektiert werden. Diese Funktion sorgt für die Freigabe des durch den Ausschnitt belegten Speicherplatzes und schafft Ordnung bei allzu vielen Pinseln.

11.3 bis 11.6 Keinesfalls benutzen
Diese Funktionen sind noch nicht implementiert. Achtung! Bei Klick auf diese Gadgets kann es zum Absturz des Rechners kommen.

11.7 Ausschnitt speichern
Der gerade angezeigte Ausschnitt wird als IFF-Bild auf Diskette abgelegt. Dazu sind Pfad und Name in einem üblichen Datei-Requester einzutragen.

11.8 Ausschnitt laden
Ein Ausschnitt wird als Pinsel geladen.

12. Spezialeffekte
Sollte dieses Menü einmal implementiert werden, finden hier solche Funktionen wie Spiegelungen, Drehungen und weitere Spielereien ihren Platz.

13. Löschen
Löscht den Zeichenbereich ganz oder teilweise. Zum Löschen des gesamten Bereiches muß nur [ALL] angeklickt werden. Ansonsten werden Bereiche einfach durch Markierung mit dem inzwischen wohlbekannten Rechteck gelöscht.

14. Farbeinstellungen
Hier kann jede Farbe, die am oberen Bildschirmrand dargestellt ist, angewählt und in ihren Werten für die Grundfarben Rot, Grün und Blau geändert werden. Dazu sind lediglich die entsprechenden Rollbalken zu bewegen.

15. Ohne Funktion
In der Version 1.21 ist das gesamte Menü 15 noch nicht mit Funktionen belegt.

16. Drucken
Alle Funktionen zur Ausgabe eines Bildes auf den Drucker, finden sich in diesem Menü.

16.1 Teilausdruck
Druckt ein Rechteck aus. Normalerweise entspricht dieses Rechteck dem gerade sichtbaren Bereich der Zeichnung. Durch Setzen eines eigenen Ausschnittes, kann jedoch jeder beliebige Teil ausgedruckt werden.

16.2 Gesamtausdruck
Gibt die gesamte SuperBitmap auf den Drucker aus.

16.3 Automatische Skalierung
Ist diese Option angewählt, drucken die Routinen des `printer.device` das Bild mit demselben Breiten/Höhenverhältnis, wie das des Bildschirmes. Alle folgenden Einstellungen werden dann ignoriert.

16.4 Mehrseitiger Ausdruck
Zur Ausgabe eines Kunstwerkes auf mehreren Seiten. Klappt am besten, wenn der Größe der SuperBitmap die Auflösung des Druckers zugrundeliegt.

16.5 - 16.14 Skalierung des Ausdruckes
Getrennt für `X` und `Y` kann der Ausdruck eines kleinen Bildes vergrößert werden.

16.15 Druckvorgang starten
Sind alle Einstellungen gemacht, startet ein Klick auf dieses Gadget den Druckvorgang.

17. Diskettenverwaltung
Dieses Menü stellt alle Funktionen zum Laden und Abspeichern von Bildern und anderen Daten bereit.

 © 1990 technicSupport

17.1 Speichern eines Bildes
Nachdem der Name des Bildes in einen Datei-Requester eingetragen worden ist, wird das Bild gespeichert.

17.2 Laden eines Bildes
Nach Auswahl in einem Datei-Requester wird ein Bild von Diskette geladen.

17.3 Packed
Die Daten werden in gepackter Form auf Diskette gebannt. Das spart Diskettenplatz und bei den meisten Programmen auch Ladezeit.

17.4 Unpacked
Das Bild wird im ungepackten Zustand gespeichert. Solche Bilder lassen sich mit eigenen Routinen leichter wieder laden.

17.5 Ausschnitt laden/speichern
Beim Speichern wird nur der eingerahmte Teil gespeichert, auch beim Laden bezieht sich die Routine nur auf diesen Ausschnitt.

17.6 Alles laden/speichern
Das gesamte Bild wird gespeichert bzw. geladen.

17.7 Farben mit abspeichern
Die sogenannte *Colormap*, üblicherweise Bestandteil eines IFF-Bildes, wird gemeinsam mit dem Bild abgespeichert beziehungsweise eingeladen.

17.8 Farben ignorieren
Die Speicher- und Laderoutinen ignorieren sämtliche Farbinformationen in den Dateien.

18. Ohne Funktion
Auch dieses Menü ist noch nicht mit Funktionen versehen.

Damit sind alle Funktionen erläutert. Ich kann Ihnen jetzt nur noch viel Spaß mit diesem Programm wünschen. Doch da gibt es im glei-

chen Verzeichnis noch einige interessante Hilfsprogramme, von denen nur die ersten beiden etwas mit `VDraw` zu tun haben.

Vier kleine Utilities

Der *Fill Editor* `Vpad` erlaubt ein einigermaßen komfortables Editieren der Füll-Muster für `VDraw` aus der Datei „VDraw:VDraw.patterns". [Load] beziehungsweise [Save] laden und speichern die Muster im IFF-Format. [Solid] füllt das Musterfeld komplett aus, [Clear] löscht es vollkommen. Mit [+] und [-] kann die Höhe des Musters beeinflußt werden. [Copy] erlaubt das Übertragen eines Musters auf andere Felder. Dazu ist das zu kopierende Muster erst anzuwählen, so daß es in der linken unteren Ecke erscheint und auch verändert werden kann. Nach Klick auf [Copy] ist das Zielrechteck für dieses Muster anzugeben. Das Muster kann mehrfach kopiert werden. Um diesen Modus zu verlassen, klicken Sie einfach auf ein blaues Feld innerhalb dieses Fensters.

Der *Brush Editor* `Vbrush` besitzt fast die gleichen Fähigkeiten wie der Fill-Editor, bezieht sich aber meist auf die Datei „VDraw:-VDraw.brushes". Außerdem ist der Zeichenbereich etwas größer, läßt aber keine vertikale Korrektur zu.

Das Programm `Vgrab` öffnet ein kleines Fenster auf dem Workbench-Bildschirm und zeigt darin alle momentan vorhandenen Intuition-Bildschirme an. Davon kann einer ausgewählt und als IFF-Bild abgespeichert werden. [List] steht für das Anzeigen einer neuen Liste, [Help] gibt einen Hilfstext aus, und [Grab] „schnappt" sich das Bild und speichert es ab.

`Vpalette` erlaubt das Ändern jeder Farbe auf einem Intuition-Bildschirm. Die Farbe wird einfach angeklickt und dann mit den Reglern verändert. [OK] beläßt die Änderungen und übergibt sie Intuition. [?] gibt einen Hilfstext aus, [S] läßt das Utility auf einen anderen Bildschirm springen, um dort die Farben zu verändern. [NO] schließlich macht alle Änderungen rückgängig. Das geht jedoch nur, solange das Fenster noch auf dem Bildschirm ist, auf dem etwas verändert wurde.

 © 1990 technicSupport

PROBLEME

Verwenden Sie keine Superbitmaps mit mehr als 1024 Punkten, sonst kann es bei einigen Programmfunktionen zum Absturz des Systems kommen. Einige Programmfunktionen sind leider noch nicht implementiert. Umständlich ist auch, daß `VDraw` nach einem Ändern wichtiger Werte jedesmal neu gestartet werden muß.

HERKUNFT

Der Autor, STEPHEN VERMEULEN, behielt sich alle Copyright-Rechte an diesem Programm vor. Trotzdem gab er es als Shareware heraus und ließ es auf Panorama 19a veröffentlichen. Als Betrag für die Programmierer-Mühen wünscht sich der Autor $20. Dies ist übrigens die letzte Shareware-Version des Programmes `Vdraw`. Mit zahlreichen Erweiterungen und Verbesserungen versehen, wird es seit einiger Zeit unter dem Namen `Express Paint` als kommerzielles Produkt vertrieben. Ein weiterer Beweis für die hohe Qualität der Public Domain! Hier die Adresse des Autors:

```
Stephen Vermeulen
3635 Utah Dr., N.W.
Calgary, Alberta
CANADA, T2N 4A6
```

VERWEISE

Zu nennen sind hier einige einfache Malprogramme, die `Vdraw` jedoch weit unterlegen sind: `SimplePaint` (Fishdisk 128), `LilPaint` (TBAG 016), `FreeDraw` (CC 031) sowie `CAO 3D` (RPD 016).

VirusX

Automatischer Virusdetektor

AUFRUF

VirusX wird aus dem CLI mit dem Kommando

`VirusX` [⟨*X*⟩ ⟨*Y*⟩] [check]

als Hintergrundtask gestartet. Der `RunBack`-Befehl ist nicht erforderlich.

OPTIONEN

`check` Diese Option bewirkt, daß `VirusX` kurz erscheint, Speicher sowie eingelegte Disketten überprüft und dann wieder verschwindet.

⟨*X*⟩ Bestimmt die X-Koordinate des `VirusX`-Fensters relativ zum linken Rand des Screens.

⟨*Y*⟩ Bestimmt die Y-Koordinate des `VirusX`-Fensters relativ zum oberen Rand des Screens. Der Befehl `VirusX 330 0` sorgt zum Beispiel dafür, daß das Fenster in der rechten oberen Ecke des Bildschirms erscheint.

ZWECK

`VirusX` durchsucht nach jedem Neueinlegen einer Diskette deren Bootblöcke nach Hinweisen auf die Existenz von Virusprogrammen. Zusätzlich wird der Speicher des Rechners überprüft.

BESCHREIBUNG

Um das Programm ständig greifbar zu haben, sollte es in das `c`-Verzeichnis der Arbeitsworkbench kopiert werden. Nach dem Aufruf, der auch innerhalb der `startup-sequence` erfolgen kann, erscheint ein schmales Fenster mit einem Schließ-Gadget, über welches das Programm wieder abgeschaltet werden kann.

© 1990 technicSupport

Aktiviert man das Fenster durch Anklicken mit der linken Maustaste und betätigt anschließend den rechten Mausknopf, so wird das Fenster größer und zeigt eine Liste der Viren, die das Programm erkennt sowie die Anzahl der geprüften und der Disketten, bei denen das Programm den Bootblock neu „installiert" hat. `VirusX` erkennt zur Zeit 16 verschiedene Viren: `SCA`, `Byte Bandit`, `North Star`, `Byte Warrior`, `Revenge`, `Obelisk`, `IRQ`, `Pentagon Circle`, `H.C.S.`, `DiskDoc`, `Graffiti`, `16 Bit Crew`, `Phantasmumble`, `Old Northstar`, `Ultra Fox` und `Lamer Exterminator`.

Entdeckt `VirusX` eines dieser Viren, so erscheint ein Requester, der den Anwender über sein Problem informiert, zum Beispiel:

```
Danger: The disk in DF<x>: is
infected with an SCA Virus!!
```

Es kann aber auch ein Hinweis erfolgen, daß ein Virus im Speicher saß und bereits entfernt worden ist.

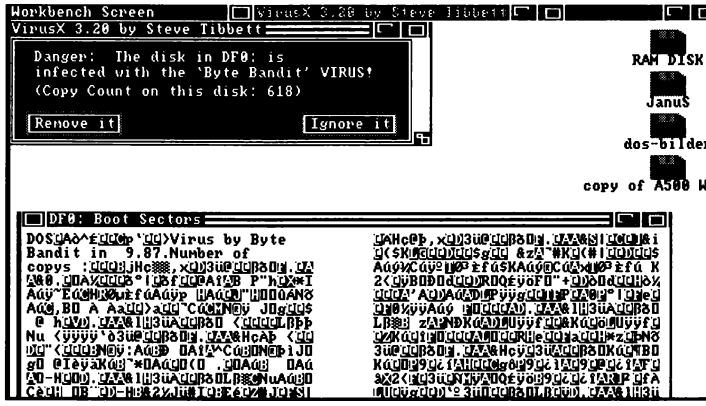
```
.... in memory, and is now disabled.
```

Weicht der Bootblock der Diskette vom Standardformat ab, ohne daß er einem der bekannten Viren zugeordnet werden kann, erfolgt eine Meldung, daß die Diskette keinen Standard-Bootblock besitzt:

```
Danger: The disk in DF<x>: has
Nonstandard Boot Code!
```

Diese Meldung kann auf ein unbekanntes Virus hinweisen, sie erscheint aber auch dann, wenn eine Diskette einen nicht üblichen, aber dennoch harmlosen Bootblock besitzt, oder wenn der Bootblock mit einem Protektor gegen Virenbefall geschützt wurde. In diesem Fall sollte man sich den Bootblock etwas näher ansehen, indem man das `VirusX`-Fenster aktiviert und anschließend die Zifferntaste drückt, die der Nummer des verdächtigen Laufwerkes entspricht. `VirusX` ändert die Gestalt des Fensters und zeigt den Inhalt der Bootblöcke an. Mit etwas Glück findet man einen lesbaren Hinweis auf ein Virus oder eine andere Funktion des Bootblockes.

Mit jeder Warnung gibt Ihnen `VirusX` die Möglichkeit, den verdächtigen Bootblock zu überschreiben, was man auch tun sollte, wenn ein Virus identifiziert worden ist. Ein nicht nicht eindeutig als Virus



Wenn Gefahr droht, schlägt VirusX Alarm

identifizierter Bootblock muß nicht automatisch gefährlich sein, er kann z. B. einen Vorspann laden oder einen Kopierschutz enthalten; diese Funktionen würden durch das Überschreiben zerstört werden. Es kann sich aber auch um ein Virus handeln, das dem Programm noch nicht bekannt ist. Die auf Amiga verbreiteten Viren lassen sich leider immer schwieriger als solche erkennen. In Zweifelsfällen sollte man den dubiosen Bootblock mit einem Programm wie `SaveBoot` oder `BootBack` archivieren, bevor man ihn überschreibt. Die auf Amiga verbreiteten Viren lassen sich leider immer schwieriger als solche erkennen.

Um einen Bootblock zu überschreiben, klicken Sie mit dem Mouse-Pointer auf den [Remove it] -Button innerhalb des Requesters und beantworten Sie die folgende Sicherheitsabfrage mit [Yes] . Über den [Ignore it] -Button können Sie den Requester entfernen, ohne ein Überschreiben zu veranlassen. Eine erneute Prüfung aller eingelegten Disketten läßt sich bei aktiviertem `VirusX`-Fenster durch Drücken der Taste [C] auslösen.

© 1990 technicSupport

Das IRQ-Virus, das sich nicht über den Bootblock verbreitet, sondern dadurch, daß es sich an Programme hängt, kann `VirusX` nur erkennen und es aus dem Speicher entfernen. Die Programme selbst bleiben infiziert. In der `VirusX`-Schublade der Fishdisk befindet sich jedoch ein weiteres Programm namens `KV`, das Dateien auf einen Virenbefall überprüft und das IRQ-Virus gegebenenfalls entfernt. Auf eine verdächtige Datei ist dazu der Befehl

`KV` ⟨*Dateiname*⟩

anzuwenden. Um alle im `C:`-Verzeichnis liegenden Programme zu überprüfen, läßt sich der UNIX-Joker `*` benutzen:

```
KV C:*
```

PROBLEME

`VirusX` wird mit Sourcecode verbreitet. Es ist daher nicht ausgeschlossen, daß Programmierer künftiger Viren Wege finden, den Prüfmechanismus dieses Programms zu umgehen.

HERKUNFT

`VirusX` wird regelmäßig dann verbessert, wenn der Programmautor Informationen über neuartige Viren erhält, so daß man auf das Erscheinen neuer Versionen achten sollte. Die hier besprochene Version 3.20 befindet sich auf Fishdisk 216. Autor ist:

```
Steve Tibbett, 2710 Saratoga Pl. #1108
Gloucester, Ontario, K1T 1Z2
```

WBLace

Schaltet die Workbench auf Interlace

AUFRUF

`WBLace` wird von der Workbench aus durch Anklicken des Icons gestartet.

ZWECK

`WBLace` erleichtert das Umschalten zwischen normaler Auflösung und Interlace-Darstellung. Wenn man mit sehr vielen Fenstern arbeitet oder im CLI große Directories bearbeitet, kann es bisweilen nützlich sein, die maximale Auflösung — den *Interlace*-Modus — zu wählen, auch wenn man dabei das störende Flimmern in Kauf nehmen muß. Es gibt auch Programme, wie beispielsweise RIM von Fishdisk 143, die nur im Interlace-Modus arbeiten können.

Commodore hat für dieses Wechseln der Auflösung einen ausgesprochen mühsamen Weg vorgesehen. Man muß die `Preferences` aufrufen, dort die Interlace-Einstellung wählen, diese Einstellung mit [Save] abspeichern und neu booten. Vielen Dank auch! Mit `WBLace` genügt es, ein Icon anzuklicken, und man ist im Interlace, nochmal klicken, und alles ist wieder beim alten.

PROBLEME

`WBLace` kann nicht funktionieren, wenn auf der Workbench Fenster geöffnet sind, die mit einem laufenden Programm assoziiert sind, z.B. ein Fenster von `VirusX` oder ein CLI-Fenster, weshalb sich das Programm auch nicht aus dem CLI heraus starten läßt. Das Schließen und Wiederöffnen solcher Fenster ist aber immer noch wesentlich schneller als ein Neustarten des gesamten Systems.

HERKUNFT

Dieses nützliche Programm findet sich auf der Panorama 018a und stammt von `Khalid Aldoseri, CIS 75166,2531`.

© 1990 technicSupport

WBLander

Kleines Spiel für den Workbench-Screen

AUFRUF

`WBLander` kann aus dem CLI heraus durch den Befehl `lander` oder durch Anklicken des Icons von der Workbench aus gestartet werden.

ZWECK

`WBLander` ist ein einfaches Geschicklichkeitsspiel, das wegen seiner Kürze auf fast jeder Diskette Platz finden kann.

BESCHREIBUNG

`WBLander` öffnet zwei kleine Fenster, die nur aus der Titelleiste bestehen. Anschließend erscheint ein Raumschiff, das mit den Cursortasten oder einem Joystick in Port 2 möglichst sanft auf einem der beiden Fenster gelandet werden muß. Das Fenster mit dem Titel „Don't land here", darf dabei nicht berührt werden. Die Landung muß dabei in der Art erfolgen, daß nicht nur die Fallgeschwindigkeit mit der [↑]-Taste auf ein Minimum reduziert wird, auch die Bewegung in horizontaler Richtung darf nicht zu groß sein.

`WBLander` ist gut geeignet, wenn man sich zwischen ernsthafteren Arbeiten einmal kurz entspannen, aber nicht gleich ein aufwendigeres Spiel laden möchte. Die besprochene Version verfügt über vielfältige Sound-Effekte und ein Feindschiff, mit dem man nicht kollidieren darf, was die Aufgabe noch etwas erschwert. Für die ersten Landeversuche kann die Schwierigkeit jedoch etwas verringert werden, indem beide Fenster — das erlaubte sowie das verbotene — mit der Maus etwas weiter voneinander entfernt werden.

HERKUNFT

Die besprochene Version befindet sich auf der Panorama 024a. Das Programm stammt von PETER DA SILVA, die Geräuscheffekte von KARL LEHENBAUER.

© 1990 technicSupport

XICON

Startet DOS-Kommandos von der Workbench

AUFRUF

Nachdem alle nötigen Vorbereitungen getroffen sind, genügt ein Doppelklick auf ein Icon, um ein Programm oder eine Batchdatei zu starten. Dabei wird `XICON` automatisch mitaufgerufen.

ZWECK

`XICON` ermöglicht das Starten von CLI-Befehlen, Batchdateien oder Programmen über das Anklicken von Icons.

BESCHREIBUNG

Batchdateien oder Programme, denen beim Aufruf eine Reihe von Parametern übergeben werden muß, lassen sich nicht ohne weiteres von der Workbench aus starten. Mit `XICON` steht Ihnen ein Hilfsprogramm zur Verfügung, das solcherlei Vorhaben auf vielfältige Weise unterstützt. Das Programm selbst kann dabei in einem beliebigen Verzeichnis stehen, dieses Verzeichnis muß jedoch in der `.info`-Datei, die alle für das Icon nötigen Informationen enthält, spezifiziert sein.

Wir brauchen uns ohnehin nicht mehr groß um das `XICON`-Programm selbst zu kümmern, da die entscheidenden Arbeitsschritte mit Hilfe des `Info`-Utilities aus dem `Workbench`-Menü vorzunehmen sind. Hier werden die nötigen Einstellungen vorgenommen, die `XICON` dann aus der jeweiligen `.info`-Datei übernimmt. Wie Sie sicherlich wissen, gibt es verschiedene Icontypen. Es ist ja auch leicht einzusehen, daß ein Icon, welches beim Anklicken eine Schublade öffnet, anders aufgebaut sein muß als eines, welches ein Programm startet. `XICON` arbeitet nur mit solchen Icons, die vom Typ „PROJECT“ sind. Icons dieses Typs finden Sie z.B. in der `XICON`-Schublade von Fishdisk 102. Sie können als Ausgangsmaterial dienen und bei Bedarf mit `IconEd` ein neues Aussehen verpaßt bekommen. Auch gibt es in der Public Domain viele Utilities, die das

 © 1990 technicSupport

Erstellen von Icons erleichtern. So lassen sich beispielsweise mit `ZapIcon` IFF-Bilder in „TOOL"-Icons umwandeln, die dann mit dem Programm `IconType` in „PROJECT"-Icons geändert werden können. Diese beiden Programme befinden sich ebenfalls auf Fishdisk 102 (im Verzeichnis „Xicon/Icon_Stuff"); eine Kurzbeschreibung finden Sie an anderer Stelle in diesem Buch.

Das Arbeiten mit `XICON` ist eine ziemliche abstrakte Angelegenheit. Aus diesem Grunde möchte ich die Einsatzmöglichkeiten zumeist an ganz konkreten Beispielen aufzeigen. Diese Beispiele werden aus Gründen der Überschaubarkeit sehr einfach gewählt sein, so daß der Aufwand für das Starten von der Workbench teilweise unangemessen hoch erscheinen kann. Da diese Beispiele experimentellen Charakter tragen, werden Sie praktischerweise in der RAM-Disk ausgeführt. So können Sie sich mit `XICON` vertraut machen, bevor Sie das Programm für konkrete Anwendungen heranziehen.

Aufrufen einer Batchdatei

Eine *Batchdatei* (Stapelverarbeitungsdatei) ist eine Datei, die eine Folge von AmigaDOS-Kommandos enthält, die über den Befehl

```
execute <Batchdatei>
```

automatisch Schritt für Schritt abgearbeitet werden. Bekanntestes Beispiel dürfte die `startup-sequence` sein, die sich im `s`-Directory befindet und nach jedem Booten für die Installation der gewünschten Arbeitsumgebung (gewöhnlich der Workbench) sorgt.

Für unsere Versuche ist grundsätzlich jede Batchdatei geeignet — sofern sie nicht gerade die RAM-Disk löscht. Ich schlage eine Datei vor, die die Befehle des `c`-Directories in die RAM-Disk kopiert und anschließend ein CLI-Fenster aufruft. Rufen Sie den Systemeditor mit

```
ed ram:Testdatei
```

auf und erzeugen Sie folgende Zeilen:

```
copy c: ram:
assign c: ram:
system:CLI
```

Anschließend verlassen Sie den Editor, indem Sie [ESC], nach dem Erscheinen des Sternchens [X] und [Return] drücken. Die Funktionsfähigkeit dieser Datei können Sie vom CLI aus mit

```
execute ram:Testdatei
```

überprüfen. Diese Batchdatei muß nun zunächst ein „PROJECT"-Icon erhalten, wozu wir das „democommand"-Icon aus der XICON-Schublade nehmen:

```
copy AmigaLibDisk102:Xicon/democommand.info ram:Testdatei.info
```

Den letzten Schliff müssen Sie diesem Icon von der Workbench aus geben. Aktivieren Sie es, indem Sie es einmal anklicken, und wählen Sie Info aus dem Workbench-Menü an. Es erscheint ein neues Fenster, das Ihnen verschiedene Möglichkeiten bietet, .info-Dateien (sprich: Icons) zu verändern. Momentan soll uns nur das Eingabefeld für das Default Tool interessieren. Dieses muß die Pfadangabe und den Namen des aufzurufenden Programms enthalten. Klicken Sie mit dem Mouse-Pointer in dieses Feld, damit der Cursor erscheint, tragen Sie in unserem konkreten Fall

```
AmigaLibDisk102:Xicon/Xicon
```

oder, wenn Sie für die Fishdisk kein freies Laufwerk haben,

```
ram:xicon
```

ein und verlassen Sie das Fenster durch Anklicken des [Save]-Buttons in der linken unteren Ecke. Die Datei „Testdatei.info" wird nun in ihrer neuen Form in die RAM-Disk geschrieben. Wenn Sie die Angabe „ram:xicon" gewählt haben, muß XICON noch mit

```
copy AmigaLibDisk102:Xicon/Xicon ram:
```

dorthin befördert werden. Jetzt genügt ein Doppelklick auf das Testdatei-Icon, um die Batchdatei abarbeiten zu lassen. XICON öffnet zunächst ein Fenster mit dem Titel „XICON 2.00", in das alle Ausgaben geleitet werden, in unserem Fall eine Auflistung der Befehle, die erfolgreich in die RAM-Disk kopiert wurden. Kurz darauf öffnet sich ein CLI-Fenster in der gewohnten Größe, welches Sie mit dem endcli-Befehl beenden müssen, um auch das XICON-Fenster wieder schließen zu können.

 © 1990 technicSupport

Maßschneidern von Icons

Über diese Grundfunktion hinaus stellt XICON viele weitere Optionen bereit, die das Programm zu einem wirklich universell einsetzbaren Hilfsmittel für die Entwicklung workbenchgesteuerter Programme machen. All diese Optionen müssen dem TOOL TYPES-Eingabefeld des Info-Programms übergeben werden. Um eine neue Option zu übergeben, klicken Sie auf den [ADD] -Button des Info-Fensters und daraufhin direkt in das Eingabefeld. Sobald der Cursor erscheint, können Sie die gewünschte Funktion eingeben. Durch die vorhandenen Funktionen können Sie sich über die Buttons mit dem aufwärts- bzw. abwärtsgerichteten Pfeil bewegen. Nicht benötigte Funktionen lassen sich überschreiben oder durch Anklicken des [DEL] -Buttons löschen. Ein kleiner Tip noch: Alle Optionen sind *unbedingt in Großbuchstaben* einzugeben, da sie andernfalls vom Programm ignoriert werden.

WINDOW=⟨*x*⟩⟨*y*⟩⟨*Breite*⟩ ⟨*Höhe*⟩[Titel]
: Größe (und Titel) des Ausgabefensters werden gemäß der AmigaDOS-Standardkonvention definiert.

TITLE=⟨*Name*⟩
: Der Titel des Ausgabefensters wird geändert.

MODE=nowindow
: Das Ausgabefenster wird unterdrückt.

MODE=closewindow
: Das Ausgabefenster schließt sich automatisch nach der Ausführung des letzten Befehls.

MODE=text
: Die Batchdatei wird als Text ausgegeben, jedoch nicht abgearbeitet.

MODE=noscript
: Die Abarbeitung der Batchdatei wird unterdrückt. Dennoch muß sie als „Dummy“ vorhanden sein. Diese Einstellung ist nur in Verbindung mit der CMD-Option sinnvoll.

CMD=⟨*Befehlszeile*⟩
: Führt eine im TOOL TYPES-Eingabefeld spezifizierte Befehlszeile unmittelbar aus, z.B.: CMD=dir opt a

ABORT-CMD=...	Wie CMD, wird jedoch nur bei gesetztem Abbruch-Signal ausgeführt.
TEXT=⟨*Dateiname*⟩	Gibt eine beliebige Textdatei aus.
ABORT-TEXT=...	Wie TEXT, wird jedoch nur bei gesetztem Abbruch-Signal ausgeführt.
SCRIPT=⟨*Batchdatei*⟩	Veranlaßt die Abarbeitung einer Batchdatei, noch bevor die eigentliche, dem Icon zugeordnete Batchdatei ausgeführt wird.
ABORT-SCRIPT=...	Wie SCRIPT, wird jedoch nur bei gesetztem Abbruch-Signal ausgeführt.
PAUSE	Unterbricht die Abarbeitung bis zum nächsten Tastendruck.
ABORT-PAUSE	Wie PAUSE, wird jedoch nur bei gesetztem Abbruch-Signal ausgeführt.
REM=⟨*Textzeile*⟩	Gibt eine im TOOL TYPES-Eingabefeld spezifizierte Textzeile aus.
ABORT-REM=...	Wie REM, wird jedoch nur bei gesetztem Abbruch-Signal ausgeführt.
LOCDIR=⟨*Dateiname*⟩	Legt in einer Datei die Angabe des Verzeichnisses ab, aus dem heraus der Aufruf von XICON erfolgte.
EXISTS=⟨*Dateiname*⟩	Setzt ein Abbruch-Signal, falls die angegebene Datei nicht gefunden wird, wodurch die Ausführung aller gewöhnlichen Befehle verhindert wird. Dasselbe Abbruch-Signal wird gesetzt, wenn das Schließ-Gadget des XICON-Fensters angeklickt wird, bevor XICON seine Arbeit beendet hat.
RESTORE	Hebt ein gesetztes Abbruch-Signal wieder auf.

© 1990 technicSupport

Nach dem Aufruf berücksichtigt XICON zuerst die *globalen* Optionen MODE=, WINDOW= und LOCDIR=, egal an welcher Position sie unter den Tool-Types-Angaben stehen. Alle anderen Optionen werden in der Reihenfolge Ihres Auftretens abgearbeitet.

Die Option MODE= darf grundsätzlich nur einmal als Tool-Type erscheinen, mehrere Einstellungen lassen sich aber unter Verwendung des „|"-Zeichens gleichzeitig realisieren, z.B.

```
MODE=nowindow|noscript
```

Aus einem einzigen Icon heraus lassen sich mehrere CMD- oder SCRIPT-Befehle ausführen, jedoch wird jeder von einem separaten CLI-Task ausgeführt, weshalb jedesmal die komplette Pfadangabe nötig wird. Für diese Pfadangabe kann Ihnen die LOCDIR-Option einen Orientierungspunkt liefern, da sie in einer beliebig wählbaren Datei eine Information über *das* Verzeichnis ablegt, in dem sich das Icon befindet. Innerhalb Ihrer Batchdatei läßt sich mit dem Befehl

```
cd <⟨Dateiname⟩ >nil: ?
```

in exakt dieses Verzeichnis wechseln. Von diesem Directory ausgehend, können Sie dann gezielt auf andere Verzeichnisse zugreifen.

BEISPIEL

In der Beschreibung des Programmes PLAY hatten wir Ihnen eine Batchdatei vogestellt, mit der sich drei Musikstücke von der RPD-Diskette 107B hintereinander abspielen lassen. Mit XICON können wir nun ein Icon erstellen, welches das Abspielen dieser Batchdatei veranlaßt. Wenn Sie dieses Beispiel nachvollziehen möchten, sollten Sie sich eine Arbeitskopie von der RPD 107B anfertigen. Diese Diskette ist nämlich randvoll, so daß es nötig wird, einige Daten zu löschen, am besten das für unser Vorhaben nicht benötigte Verzeichnis „Emp_Instruments". Legen Sie dazu Ihre Arbeitskopie ein und geben Sie vom CLI aus den Befehl:

```
delete CAM_107b:Emp_Instruments all
```

Kopieren Sie jetzt die Programme XICON, PLAY (von RPD 107A) sowie das Icon democommand.info in das Basisverzeichnis der Arbeits-

kopie. Die in der PLAY-Programmbeschreibung vorgestellte Batchdatei können Sie mit Notepad oder dem Systemeditor ed erstellen und unter einem beliebigen Namen — wiederum in das Basisverzeichnis der Arbeitskopie — abspeichern. Ich wähle hier den Namen „Spieldrauflos". Diesen Namen, ergänzt um das Suffix „.info", muß auch das Icon tragen. Verwenden Sie zum Umbenennen den Rename-Befehl aus dem Workbench-Menü oder führen Sie die Namensänderung vom CLI aus durch:

```
rename democommand.info as Spieldrauflos.info
```

Nun können Sie das CLI verlassen, auf der Workbench das Spieldrauflos-Icon *einmal* anklicken und das Info-Hilfsprogramm aufrufen, um die letzten Einstellungen vorzunehmen. Klicken Sie in das DEFAULT TOOL-Eingabefeld und geben Sie dort

```
CAM_107B:xicon
```

ein. Anschließend klicken Sie auf den [ADD] -Button und in das TOOL TYPES-Eingabefeld, wo Sie als erste Eintragung

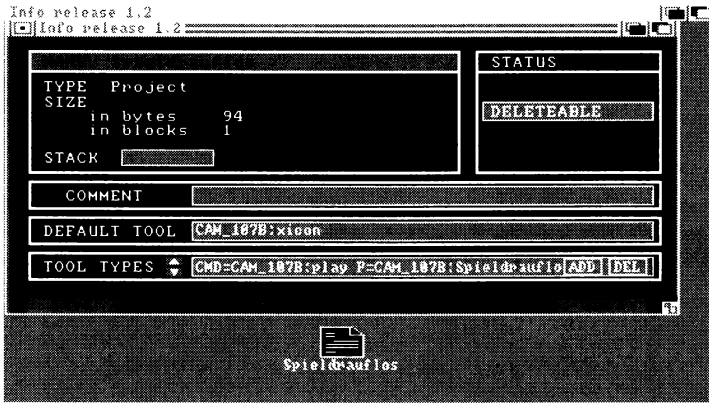
```
MODE=nowindow|noscript
```

vornehmen. Die Option nowindow unterdrückt jede Ausgabe, also auch Fehlermeldungen. Für die Erprobungsphase neuer Icons kann es daher nützlich sein, auf diese Option zu verzichten. Unbedingt notwendig ist dagegen der Befehl noscript, da unsere Befehlsdatei keine für AmigaDOS verständlichen Informationen enthält und daher auch nicht abgearbeitet werden darf. Den eigentlichen Programmaufruf realisieren wir mit der CMD-Option, indem wir nochmals den [ADD] -Button anklicken und in das nun wieder leere TOOL TYPES-Eingabefenster

```
CMD=CAM_107B:play P=CAM_107B:Spieldrauflos
```

eintragen. Nach dem Verlassen des Info-Fensters über den [Save] -Button kann „Spieldrauflos" mit einem Doppelklick gestartet werden.

 © 1990 technicSupport



Das Info-Tool: Schlüssel für Icon-Manipulationen

PROBLEME

Wenn `XICON` nicht funktioniert, kann es daran liegen, daß sich der AmigaDOS-Befehl `run` nicht im `C:`-Verzeichnis befindet. Zu beachten ist hier der Doppelpunkt. Gemeint ist nicht das `c`-Verzeichnis irgendeiner Diskette, sondern das logische Verzeichnis `C:`, in dem Amiga seine DOS-Befehle sucht. Der genaue Name dieses Verzeichnisses läßt sich mit `assign` in Erfahrung bringen. Sollte sich in diesem Verzeichnis kein `run`-Befehl finden, so muß er von der Original-Workbenchdiskette dorthin kopiert werden.

Das Ausgabefenster von `XICON` akzeptiert keine Benutzereingaben von der Tastatur, da der „Eingabekanal" schon für die Batchdatei reserviert ist. Die einfachste Lösung dieses Problems besteht in der Erzeugung zweier Batchdateien, von denen die erste, die über `XICON` aufgerufen wird, lediglich ein neues (uneingeschränkt arbeitendes) Fenster für die zweite Datei aufruft, z.B. über

`newcli con:10/10/200/100/AuchEingabe from` ⟨*Batchdatei2*⟩

In diesem Fenster werden nun automatisch die in Batchdatei2 enthaltenen Befehle ausgeführt, wobei auch Tastatureingaben (z.B. bei Verwendung des `ask`-Befehls) zugelassen sind.

HERKUNFT

Die hier gegebene Programmbeschreibung trifft auf die Versionen 2.00 (Fishdisk 102) und 2.01 (Fishdisk 157) zu. Auf der Fishdisk 102 befinden sich neben dem Programm einige Beispiele sowie Hilfsprogramme, die auf Fishdisk 157 fehlen. Diese enthält jedoch eine neuere Programmversion, bei der ein Fehler hinsichtlich der `LOCDIR`-Funktion beseitigt wurde, sowie den auf Nr. 102 vermißten Quellcode `Findwindow.c`.

Der Autor überläßt es Ihrer Entscheidung, ob Ihnen das Programm $5 oder $10 wert ist. Seine Anschrift lautet:

```
Pete Goodeve
3012 Deakin Street #D
Berkeley, Calif. 94705
USA
```

VERWEISE

`IconExec` (Fishdisk 012), `CLIRun` (RPD 056), `IconX` von der Workbench 1.3

© 1990 technicSupport

ZapIcon

Generiert Icons aus IFF-Bildern

AUFRUF

`ZapIcon` läßt sich nur aus dem CLI heraus aufrufen:

`ZapIcon` ⟨*IFF-Bild*⟩ ⟨*Icon*⟩

OPTIONEN

`IFF-Bild` Der Name der IFF-Bildvorlage.

`Icon` Der Name des zu erzeugenden Icons ohne das Suffix „.info".

BESCHREIBUNG

`ZapIcon` ist ein kleines Utility, welches beliebige IFF-Bilder in Icons verwandelt. Solche Bildvorlagen lassen sich unter anderem mit `DeluxePaint` erstellen und als Brush (Pinsel) abspeichern. Hierbei läßt sich die Größe dieser Brushes beliebig wählen. Die verwendete Farbpalette sollte sich allerdings auf die 4 Farben beschränken, die Sie in den Preferences für Ihren Workbench-Screen festgelegt haben, damit sich die ursprünglichen Bilder auch in den Icons noch wiedererkennen lassen.

`ZapIcon` erzeugt ausschließlich Icons vom Typ „TOOL". Diese lassen sich aber mit anderen Hilfsprogrammen wie z.B. `IconType` leicht in andere Icontypen umwandeln.

HERKUNFT

`ZapIcon` befindet sich im Unterverzeichnis „graphic" der Panorama-Diskette 11. Es ist ebenfalls im Unterverzeichnis „Xicon/Icon_Stuff" von Fishdisk 102 zu finden. Über den Autor dieses Programms ist nichts bekannt.

© 1990 technicSupport

Alle Fishdisks bis Nr. 282

Colorful	Demonstriert alle 4096 Farben	001
Amigademo	Grafisches Vergleichstest-Programm	001
Gfxmem	Grafische Speicheranzeige in Assembler	001
Dhrystone	Dhrystone-Vergleichstest Programm	001
Balls	Zeichnet Bälle	001
FreeDraw	Ein kleines Zeichenprogramm	001
Trackdisk	Zeigt Zugriff auf trackdisk-Treiber	001
Gad	Zeigt, wie man Gadgets erzeugt	001
Latffp	Zeigt Zugriff auf Motorola-Fließkommafunktion	001
SpeechToy	Experimente mit Sprachausgabe	001
Requesters	Zeigt Aufbau einfacher Abfrager	001
AmigaTerm	Terminalemulation mit Xmodem-Protokoll	001
Halfbrite	Führt Halbhelligkeits-Verfahren vor	001
Speech	Zeigt Sprachausgabe	001
Dotty	Vorführprogramm mit Punkten	001
Palette	Zeigt Bau einer Farbpalette	001
Hello	Zeigt Aufbau eines kleinen Fensters	001
Xrf	Erstellt Kreuzverweise von C-Programmen	002
MicroEmacs	Kleine Ausgabe des bekannten Texteditors	002
Make	Ähnlich wie das UNIX-make Programm	002
Portar	Archivprogramm	002
Dbug	Unterstützt Testen von C-Programmen	002
Make2	Ähnlich wie das UNIX-make Programm	002
Alib	Objekt Modul Bibliothekshilfe	002
Cc	Compiler-Aufrufer für Lattice C	002
Roff	Ein einfacher Textformatierer	003
Gothic	Druckt große gothische Buchstaben	003
Cforth	Die Programmiersprache FORTH	003
XLisp	Ein objekt-orientierter LISP-Dialekt	003
Ff	Ein schneller Textformatierer	003
Kermit	Einfaches Terminalprogramm	004
Banner	Gibt große Buchstaben aus	004
Bison	Ein Parser-Generator, wie UNIX-yacc	004
MyCLI	Erweitert das CLI, bessere Funktionen	004
Mandel	Die Mandelbrotmenge wird berechnet	004
Grep	Sucht Wort in einer Datei	004
Bm	Sucht Wort in einer Datei	004
Bgrep	Sucht Wort in einer Datei	004
Print.support	Unterstützende Programme für Druckerzugriff	005
Region	Zeigt Unterteilung in Rechteck-Regionen	005
Proctest	Beispiel zum Erzeugen eines Prozesses	005

Printer	Zeigt Zugriff auf Drucker	005
FreeMap	Zeigt freien Speicher an	005
Timer	Zeigt Zugriff auf die timer.device	005
Text.demo	Demonstriert Zugriff auf fonts	005
Mandelbrot	Visualisiert die Mandelbrot-Menge	005
Trackdisk	Zeigt Zugriff auf trackdisk.device	005
Speech.demo	Einfaches Sprachausgabe-Programm	005
Layers	Zeigt Benutzung der layers.library	005
Mouse	Zeigt Abfrage der zweiten Mausbuchse	005
Parallel	Zeigt Zugriff auf Parallel-Schnittstelle	005
Samplefont	Beispiel-Zeichensatz in Assembler	005
Serial	Zeigt Zugriff auf serielle Schnittstelle	005
One.window	Zeigt Window-Öffnung mit Console.device	005
SinglePlayfield	Zeigt Erzeugen eines single Playfields	005
Cons	Zugriff auf console.device	005
SpeechToy	Experimente mit Sprachausgabe	005
Input.dev	Veränderung des Eingabe-Stromes	005
Joystick	Zeigt Zugriff auf Joystick	005
Keyboard	Zeigt Direktzugriff auf Tastatur	005
Sortc	Sortiert eine Datei	006
Setserial	Modifiziert serielle Schnittstelle	006
Stripc	Entfernt überflüssiges aus C-Programm	006
MicroEmacs	Kleine Ausgabe des bekannten Texteditors	006
Mult	Gibt mehrfache Zeilen in einer Datei aus	006
Dadc	Analog/Digital-Uhr	006
Compress	Adaptive Lempel-Ziv Datenkomprimierung	006
Setparallel	Modifiziert Parallel-Schnittstelle	006
Scales	Zeigt vierstimmige Nutzung der audio.device	006
Hack 1.01	Das legendäre Abenteuerspiel	007
Hack	Der Quellcode zu dem berühmten Spiel	008
SetLace	Schaltet Interlace auf WB ein und aus	009
SkewB	Grafikdemo, Rubik-Würfel	009
Sparks	Grafisches Vorführprogramm mit Bewegung	009
Moire	Erzeugt Moire-Interferenz-Muster	009
Proff	Leistungsfähiger Amiga-Formatierer	009
MVP-FORTH	Implementation der Sprache FORTH	009
Trek73	Einfaches Weltraum-Strategiespiel	010
FixObj	Entfernt Überflüssiges von XModem-Dateien	010
LD	Ersatz für CLI-Dir-Befehl	010
FileZap	Erlaubt Verändern binärer Dateien	010
IFF	Programme zum Lesen und Schreiben von IFF	010
YachtC	Das bekannte Würfelspiel Yahtzee	010
Conquest	Interstellares Eroberungsspiel	010
Dehex	Wandelt hexadezimale Datei nach Binär	010
Sq.usq	Datei-Kompression und -Dekompression	010

© 1990 technicSupport

Ls	Listet Diskettenverzeichnis nach Muster	010
Pictures	Verschiedene Bilder	011
DpSlide	Diaschau-Programm	011
Amiga3d	Zeigt ein sich drehendes Amiga-Zeichen	012
StarTerm	Terminalemulator mit ASCII und XMODEM	012
Images	Verschiedene Bilder	012
LD4	Ersatz für CLI-Dir-Befehl	012
ArgoTerm	Wenig komfortables Terminalprogramm	012
Arrow3d	Zeigt einen sich schnell drehenden Pfeil	012
IconExec	Startet ein Programm von einem Icon aus	012
SetAlternate	Hilfe zum Erstellen eines Icons aus 2 Bildern	012
BASIC	Verschiedene BASIC-Programme	013
Dex	Extrahiert nroff-Quelle aus Dokumentation	014
Gi	Wandelt IFF-Brush nach C-Quellcode	014
Termcap	Termcap-terminal-Datenbank für Amiga	014
FileZap	Erlaubt Verändern binärer Dateien	014
Shell	Verbessert CLI zu einer Shell	014
Gfxmem	Grafische Speicheranzeige in Assembler	014
Beep	Eine Funktion, die einen Ton erzeugt	014
Pdterm	Terminalemulator für ANSI oder VT100	014
Dimensions	Demonstriert vierdimensionale Grafik	014
OkidataDump	Okidata ML92 Druckertreiber und Screendump	015
Polyfractals	Fractal-Programm in AbasiC	015
Monopoly	ABasiC-Brettspiel	015
Clock	Kleine Digitaluhr in der Menüleiste	015
Polydraw	Malprogramm in AbasiC	015
Fish	Zeigt schwimmenden Fisch	015
Dazzle	Kaleidoskop-Grafikdemo	015
Blobs	Buntes Grafikdemo	015
IFF	Diskette voller IFF-Informationen	016
HAM	Diskette voller HAM-Bilder mit 4096 Farben	017
Multidim	Ein 2- bis 6-dimensionaler Würfel rotiert	018
Scrimper	Druckt den Workbench-Screen aus	018
AmigaDisplay	Terminalemulator	018
XLisp 1.6	Ein objekt-orientierter LISP-Dialekt	018
PigLatin	Übersetzt Englisch in komischen Dialekt	018
Browser	Betrachten von Verzeichnissen und Dateien	018
MC68010	Information über Einbau eines 68010	018
Ash	C-Shell-ähnliche CLI-Verbesserung	018
LockMon	Zeigt Datei-Locks an	019
Keymap_Test	Programm zum Test der Keymap-Funktionen	019
JayMinerSlides	Amiga Blockdiagramme von Jay Miner persönlich	019
BlackJack	BlackJack Spiel (keine Grafik)	019
SetMouse2	Anschluß der Maus an der 2. Buchse	020
Random	Zufallszahlenerzeuger in Maschinensprache	020

Ted	Demoversion des Editors TXed	020
Hd	Hexadezimale Dateiausgabe in Prozessorformat	020
DiskSalv	Versucht, zerstörte Diskette zu reparieren	020
Pack	Entfernt überflüssige Leerzeichen aus C-Code	020
AmigaToAtari	Wandelt Amiga Objekt-Code nach Atari-Format	020
PortHandler	Ein Beispiel für einen Port-Handler	020
MultiTasking	Beispielprogramm Multitasking mit Erklärung	020
SpeechTerm	Terminalemulator mit XMODEM und Sprachausgabe	020
Hash	Berechnet Amiga-DOS-Hash-Werte	020
MandelBrots	Einige Mandelbrot-Bilder	020
MandelbrotSE	Programm zum Erforschen der Mandelbrot-Menge	021
Pemacs	Microemacs, verbessert von A. Poggio, Editor	022
Lemacs	Microemacs Version 3.6, Editor	022
Emacs v30	Version 30 von Microemacs	023
Csh	Verbessert das CLI zu einer Shell	024
Conquest	Interstellares Eroberungsspiel	024
Modula-2	Vorabversion Modula-2 von der ETH Zürich	024
Hack 1.01E	Das legendäre Abenteuerspiel	025
Ps	Zeigt Daten der laufenden Prozesse an	026
C-Kermit	Kermit, ein Datenübertragungsprogramm	026
UnHunk	Manipulation der Amiga Ladedateien	026
Archx	Bündelt Textdateien in eine Datei	026
MemExpansion	Erklärt Bau einer 1MB-Erweiterung für 1000er	027
ScienceDemos	Demo-Programme aus der Wissenschaft	027
DvorakKeymap	Beispiel einer keymap-Struktur "Dvorak"	027
Hypocycloids	Ein elektronischer Spirograph	027
ABdemos	Verschiedene ABasic Programme	027
DisAssem	Einfacher 68000-Speicherdisassembler	027
LinesDemo	Zeigt, wie man in großem Fenster rollt	027
SafeMalloc	Absicherung von malloc()-Aufrufen	027
SuperBitMap	Beispiel für ScrollLayer()-Aufruf	028
Cribbage	ABasiC-Kartenspiel	028
Shar	Packen und Entpacken von Dateien	028
Othello	Othello (Reversi), Spiel in ABasic	028
BackGammon	Backgammon in ABasic	028
MileStone	MileStone, Spiel in ABasiC	028
Cpp	Decus C Vorübersetzer	028
Rubik	Ein sich bewegender Rubikscher Würfel	029
Enough	Erlaubt, festzustellen, ob Resourcen vorhanden	029
Player	Spielt Sequenzen des Aegis Animator ab	029
Cc	MANX-Version des cc-Compiler Kontrolleurs	029
StringLib	Re-Implementation der UNIX-Bücherei	029
AegisDrawDemo	Demo-Version des bekannten CAD-Programmes	029
VT100	Terminalprogramm, VT100-Emulation	029
FineArt	Einige schöne Bilder	030

© 1990 technicSupport

FontEditor	Zeichensatz-Editor	030
MenuEditor	Erlaubt des Erstellen von Menüs	030
StarTerm	Ein Terminalprogramm	030
BBS 1.1	Mailbox-System für den Amiga	030
Life	Schnelle Implementation von Conways Life	031
TxEd	Demoversion eines Texteditors	031
MxExample	Zeigt mutual exclude gadget-Behandlung	031
Vdraw	Malprogramm	031
Mandelbrot	Erzeugt Mandelbrot-Grafiken	031
RamSpeed	Mißt rohe Speicher-Geschwindigkeit	031
Set	Ersetzt das MANX set-Kommando	031
Xicon	Erlaubt Aufruf von CLI-Scripts über Icons	031
Tree	Zeichnet rekursiven Baum	031
Input	Liest Text von Benutzer in CLI-Script ein	032
Math	Erlaubt einfache Grundrechnungen	032
Puzzle	Einfaches Quadrat Puzzle	032
Pref	Auswahl der Preference-Datei möglich	032
Sword	Text-Abenteuerspiel in AmigaBASIC	032
MacView	Erlaubt Betrachten von MacPaint-Bildern	032
Spin3	Grafikdemo mit vielen 3D-Würfeln	032
ShowHAM	Zum Anzeigen von HAM-Bildern	032
Wc	Zählt Wörter und Zeichen in ASCII-Text	032
Klondike	AmigaBASIC Spiel	032
FixEd	Entfernt Zeichen, die ED nicht verträgt	032
Canfield	AmigaBASIC Spiel	032
Asc	Gibt ASC-Code eines Zeichen aus	032
Trails	Die Maus erhält einen Schwanz, in Modula-2	032
Help	Erklärt AmigaDOS Befehle	032
Print	Kontrolliertes Ausdrucken von Textdateien	032
AreaCode	Gibt Ort in den USA nach Code aus	032
Conv	Wandelt Zahlensystem-Darstellungen um	032
Col	Stellt WB-Screen auf 60 oder 80 Zeichen	032
AutoNote	Setzt Filenote einer Datei auf ersten String	032
Timer	Zählt Zeit für Telefongespräche	032
Calendar	Kalender in AmigaBASIC	032
UnLock	Hebt Löschverbot einer Datei auf	032
Chr	Gibt Taste für gegebenen ASC-Code aus	032
Lock	Setzt Löschverbot für Dateien	032
Bit	Erlaubt Bit Operationen	032
Oing	Viele kleine springende Bälle	033
Dbuf.gels	Zeigt Bob- und Sprite-Animation	033
VT100 2.0	Terminalprogramm, VT100-Emulation	033
MemView	Zeigt direkten Ausschnitt aus dem RAM	033
Stars	Sterne fliegen auf den Betrachter zu	033
ScreenDump	Druckt obersten Bildschirm aus	033

© 1990 technicSupport

BigMap	Zeigt Arbeiten mit ScrollVPort	033
Sdb	Einfache relationale Datenbank	033
DiskMapper	Zeigt Blockbelegung einer Diskette	033
3D-Stars	Sterne fliegen auf den Betrachter zu	033
Sproing	Springende Bälle mit Klang	033
TermPlus	Variante des AmigaTerm-Terminalprogramms	033
Btree2	Ursprüngliche Version binärer Bäume	034
Less	Bequemes Betrachten von Textdateien	034
Requester	Datei-Abfrager	034
Blink	Link-Programm als Ersatz für ALink	034
Btree	Implementation von binären Bäumen	034
Browser	Erlaubt Durchstreifen von Verzeichnissen	034
NewFonts	Achtundzwanzig Amiga-Zeichensätze	034
Calendar	Verabredungskalender	034
Pr	Ausdrucken von Textdateien	034
Alint	Unterstützung für Gimpel Lint auf dem Amiga	034
VTtest	Testet VT100-Kompatibilität	035
ASendPacket	Zeigt, wie packets DOS-Handler gesendet werden	035
Plop	Kleines IFF-Lese Programm	035
ScrollPf	Rollen eines Playfields	035
SendPacket	Erlaubt Senden von AmigaDOS Packets	035
TriClops	Dreidimensionales Actionspiel	035
MacView	Erlaubt Betrachten von MacPaint-Bildern	035
Tracker	Erlaubt Fernübertragung einer Kickstart-Disk	035
DirUtil	Erlaubt CLI-Dateioperationen über die Maus	035
DirUtil 2	Erlaubt CLI-Dateioperationen über die Maus	035
Tsize	Druckt Verzeichnisgrößen einer Disk aus	035
SpriteMaker	Gemalte Sprites werden in C-Code verwandelt	035
PopCLI	Öffnen eines CLI-Fensters auf Tastendruck	035
UnIfdef	Wertet ifdefs in C-Programm aus	035
QuickCopy	Diskettenkopierprogramm	035
ConsoleWindow	Zeigt Finden des Intuition-Zeigers auf CON:	035
FileRequester	Demo eines Datei-Abfragers	035
DietAid	Hilft beim Einhalten einer Diät	036
VT100 2.2	Terminalprogramm, VT100-Emulation	036
YaBoing	Zeigt Gebrauch von Hardware-Sprites	036
KickBench	Vereinigt WB und Kickstart auf einer Diskette	036
Acp	UNIX-artiges Dateikopierprogramm	036
Vc	VisiCalc-ähnliche Tabellenkalkulation	036
Clock	Kleine Digitaluhr in der Menüleiste	036
Echo	Verbessertes Echo, mit Farbe und mehr	036
Lex	Erstellt Lesbarkeitsbewertung eines Textes	036
TunnelVision	ABasiC-Programm, 3d-Tunnel-Grafik	036
Fm	Zeigt an, welche Sektoren eine Datei belegt	036
Csh 2.01A	Verbessert das CLI zu einer Shell	036

© 1990 technicSupport

FixHunk	Korrigiert Programm für großen Speicher	036
Smalltalk	Objekt-Programmiersprache	037
NewStat	Verbesserte Version des AmigaDOS-Status	038
UUdecode	Codieren und Decodieren von Binär-Dateien	038
Jsh	CLI-Verbesserung, ähnlich BSD C-Shell	038
VoiceFiler	Stimmen-Speicher für Yamaha DX Sythesizer	038
Hp-10C	Modula-2 Programm für guten Taschenrechner	038
FixObj	Entfernt Überflüssiges von XModem-Dateien	038
Window	Zeigt Erstellung eines DOS-Fensters	038
Handler	Zeigt Erstellung eines AmigaDOS-Handlers	038
IFFDump	Gibt IFF-Datei in lesbarem Format aus	038
IFFEncode	Speichert Screen als IFF-Datei	038
Vdraw	Malprogramm	038
Reversi 6.1	Spielstarkes Brettspiel	038
Squared	Erstellt grafische Muster	038
Echo	Assembler Version von Echo	039
Cls	Bildschirmlöschen in Assembler	039
XLisp 1.7	Ein objekt-orientierter LISP-Dialekt	039
Display	Zeigt HAM-Bilder, 4096 Farben mit Beispielen	039
Driver	Beispiel für Geräte-Treiber	039
Touch	Dateidatum-Aktualisierung in Assembler	039
List	Diskverzeichnis in Assembler	039
Ask	Erlaubt Benutzerabfragen in Batchdateien	039
AreaCode	Gibt Ort in den USA nach Code aus	040
Ahost	Terminalprogramm: ANSI, B, Kermit und Xmodem	040
PagePrint	Druckt Textdateien	040
AmigaMonitor	Zeigt Systemzustand dynamisch an	040
Blink	Link-Programm als Ersatz für ALink	040
SpriteEd	Erlaubt das Editieren zweier Sprites	040
Cosmo	Asteroiden-Ballerspiel	040
PopCLI	Öffnen eines CLI-Fensters auf Tastendruck	040
DirUtil	Erlaubt CLI-Dateioperationen über die Maus	040
X-Spell	Ein Spellchecker, überprüft Rechtschreibung	040
Dg210	DG D-210 Terminalemulator	040
DosHelper 1.60	Gibt Hilfe zu AmigaDOS-Befehlen	040
Arc	Komprimierendes Dateiarchivierungsprogramm	040
DualPlayField	Beispiel eines dual playfield screens	041
LatticeXref	Erstellt Kreuzverweisliste aus C-Programm	041
Lines	Zeichnet Linien als Demo	041
Csh 2.03	Verbessert das CLI zu einer Shell	041
SetFont	Setzen eines neuen CLI-Zeichensatzes	041
AmigaVenture	AmigaBASIC Programm zum Adventure-Schreiben	041
Dbug	Unterstützt Testen von C-Programmen	041
VT100 2.3	Terminalprogramm, VT100-Emulation	041
GetFile	Dateinamen-Abfrager	041

MicroGNUEmacs	Amiga Version (MG) des bekannten Editors	042
InstIFF	Wandelt Klangdateien ins IFF-Format	043
Bbm	Demo-Version eines Geschäfts-Systemes	043
BbsList	Amiga Mailbox-Programm	043
Wild	UNIX-ähnlicher Mustervergleich	043
STEmulator	Gag: Emuliert ST-Benutzeroberfläche	043
SpriteClock	Clock als Sprite	043
WBRun	Erlaubt, Workbench-Icons vom CLI zu starten	043
BasicBoing	AmigaBasic Bewegung mit Seiten-Umschaltung	043
PopColours	Zum Verändern der Screenfarben	043
Copper	Copper-Listen-Dissasembler	043
Cc	Compiler-Kontrolleur für Manx und Lattice	043
NewIFF	Neue Informationen über Klang-IFF Dateien	044
ViewILBM	Zum Betrachten von IFF-Bildern	044
Icons	Verschiedene Icons	044
RayTracePics	Mit Ray Tracing erstellte Bilder	044
Make	Ähnlich wie das UNIX-make Programm	045
Clue	AmigaBasic-Detektivspiel (Cluedo)	045
Update	Hilfe beim Auffrischen einer Datei	045
WhereIs	Sucht auf Geräten nach einer Datei	045
Pictures	Verschiedene Bilder	045
Jive	Verwandelt Englisch in Jive-Slang	046
My.lib	Laufzeit-Bücherei, verschiedene Funktionen	046
Asm	68010 Makro-Assembler	046
Egad	Gadget-Editor, erzeugt C-Quellcode	046
CheckModem	Prüft, ob ein Modem angeschlossen ist	046
ValSpeak	Verwandelt Englisch in Val-Slang	046
ProffMacros	Makros für proff-Formatierer, wie ms und mm	046
Juggler	Jongleur-Animation, mit Sculpt 3D erstellt	047
VT100 2.4	Terminalprogramm, VT100-Emulation	047
3D-Arm	Simuliert dreidimensionalen Roboter-Arm	047
Du	Zeigt Disketten-Ausnutzung an	048
Bru	Dateiarchivierungsprogramm	048
Profiler	Bestimmt Laufzeit von Funktionen unter MANX C	048
Csh 2.04	Verbessert das CLI zu einer Shell	048
Csh 2.04M	Verbessert das CLI zu einer Shell	048
Diskperf	Disk-Vergleichstest-Programm	048
Comm 1.30	Terminalprogramm	048
MemWatch	Warnt Programme, wenn der Speicher knapp wird	048
MyUpdate	Auffrischen von C-Verzeichnissen	049
Polygon	Erzeugt Moire-Figuren	049
QMouse	Abfrage des Maus-Knopf-Zustandes in Scripts	049
Multidef	Findet mehrfach definierte Objekt-Symbole	049
Trees	Läßt einen grafischen Baum wachsen	049
Touch	Setzt Dateidatum auf aktuelles Datum	049

© 1990 technicSupport

DirUtil	Erlaubt CLI-Dateioperationen über die Maus	049
Cycloids	Spirograph-Programm	049
Plot	Plottet dreidimensionale Funktionen	049
UnixArc	Arc, ähnlich wie UNIX System V arc	050
BreakOut	Dreidimensionales Geschicklichkeitsspiel	050
Wombat 3.01	Vt-102 Emulation	050
DiskZap 1.1	Disketten-Editor	050
Asm	68010 Makro-Assembler	050
Sizzlers	Grafikdemo	050
Missile	Schnelles Actionspiel	050
FirstSiliCon	CLI-Verbesserung	050
PerfectSound	PerfectSound-Demo, mit Klang-Editor	050
Bison	Ein Parser-Generator, wie UNIX-yacc	051
Cos	AmigaBasic-Spiel mit Glücksrädern	051
Compress	Datei-Komprimierung nach Lempel-Ziv	051
Sq.usq	Datei-Kompression und -Dekompression	051
DifSsed	Dateiunterschiede werden gespeichert	051
Fractal	Erzeugt fraktale Landschaften	052
MxGads	Zeigt sich gegenseitig ausschließende Gadgets	052
Assign	Ersatz für den Amiga Assign-Befehl in C	052
Poly	Polygon-Zeichen-Vorführung	052
TEK4010	Ein Tek 4010 Terminalemulator	052
HAMPoly	HAM-Grafikdemo	052
Vdraw 1.16	Malprogramm	052
Vdraw 1.19	Gutes Malprogramm	052
FileNote	Dateikommentar, mit Assembler-Quellcode	053
Echo	Text-Ausgabe, mit Assembler-Quellcode	053
Cd	Aktuelles Verzeichnis, mit Assembler-Code	053
Break	Abbruchsteuerung, mit Assembler-Quellcode	053
UUencode	Binäre Dateien werden in Buchstaben kodiert	053
Compiler	C-Compiler, mit Quellcode in C	053
ARCre	Umbennungen für "arc"	053
TarSplit	Extraktion von Dateien aus Archiven	053
SpreadSheet	Tabellenkalkulation	053
Makedir	Neues Verzeichnis, mit Assembler-Quellcode	053
Chmod	Dateirechte, mit Assembler-Quellecode	053
Animations	Einige Aegis Animationen (mit Abspieler)	053
Ing	Gag: Hüpfende Windows	054
MoreRows	Erlaubt zusätzliche Zeilen auf Bildschirm	054
Hanoi	Demonstriert das Spiel der Türme von Hanoi	054
Tilt	Gag: Bildschirm steht schräg	054
MidiTools	Spielen und Aufnehmen für MIDI-Schnittstelle	054
Lav	Zeigt Systemauslastung an	054
ISpell	Prüft englische Rechtschreibung	054
VT100 2.06	Terminalprogramm, VT100-Emulation	055

Csh 2.05	Verbessert das CLI zu einer Shell	055
ScreenSave	Speichert Bildschirm auf Diskette ab	055
SoundExample	Erklärt Tonausgabe für Programmierer	055
Palette	Kontrolle über Farben eines Programmes	055
PipeDevice	Datentransfer zwischen zwei Programmen	055
ShanghaiDemo	Eingeschränkte Version des Spieles	055
NewStartups	1.2-Startups, für C-Programmierer	055
Vsprites	Beispiel zur VSprite-Programmierung	055
ClipBoard	Gerätetreiber für das Clipboard	056
MergeMem	Verschmilzt verschiedene Speicherbereiche	056
ConPackets	Zeigt Benutzung von DOS-Packets	056
Icon2C	Wandelt Icons nach C-Quellcode	056
GetVolume	Zeigt alle angeschlossenen Volumes	056
MCad 1.1	Sehr brauchbares CAD-Zeichenprogramm	056
GetDisks	Zeigt alle angeschlossenen Disketten	056
CutAndPaste	Ausschneiden von Teilen einer Textdatei	057
Ogre	Taktisches Spiel	057
MouseReader	Lesen von Textdateien mit der Maus	057
Splines	Zeigt Spline-Kurven mit Pop-Up-Menüs	057
GraphIt	Plottet einfache Funktionen	057
Juggler	Berühmte Animation eines Jongleures	057
EGraph	Zeichnet Kurven nach Datenpaaren	058
MemClear	Löscht freien Speicherplatz	058
View	Betrachter für ILBM-Bilder	058
WBdump	Workbench-Drucker-Treiber für Epson	058
NewZap	Dateiorientierter Diskettenmonitor	058
SmusPlayers	IFF-SMUS-Spieler	058
HyperBase	Kleine Dateiverwaltung	058
ASDG-RRD	Resetfeste RAM-Disk, Inhalt überlebt Reset	058
Rainbow	Eine Farbspielerei, manipuliert Copper-Listen	058
BigView	Zeigt auch übergroße IFF-Bilder an	058
Browser	Betrachten des Disketteninhaltes	059
DropCloth	Unterlegt Workbench mit IFF-Bild	059
Clock	Kleine Digitaluhr in der Menüleiste	059
Dme 1.22	Konfigurierbarer Texteditor	059
Browser2	Betrachten des Disketteninhaltes	059
MCad 1.2.2	Sehr brauchbares CAD-Zeichenprogramm	059
Robotroff	Gag: Sprite, das Mauszeiger hinterherjagt	059
DropShadow	Workbench Fenster erhalten einen Schatten	059
Supermort	Programm für Finanzberechnungen	059
FixWB	Erzeugt Hintergrund für Workbench	059
Blitz	Sehr schnelles Anzeigen von Text-Dateien	060
Uedit 2.0	Flexibler ASCII-Texteditor	060
PrtDrvGen	Erstellung beliebiger Druckertreiber	060
BlitzFonts	Beschleunigt Amiga-Textausgabe	060

© 1990 technicSupport

Show 2.1	Slideshow Programm	060
HandShake	VT-52, VT-100 und VT-102 Terminalemulator	060
MEd	Ein Texteditor für das CLI	060
Ueturbo	Unterstützung für den Editor ue	060
VSprites	Beispiel zur VSprite-Programmierung	061
LPatch	Ändert Programme für Workbench 1.2 ab	061
PearlFont	Ein abgerundeter Zeichensatz	061
MicroEmacs 3.8b	Editor, programmierbar	061
FillDisk	Füllt leeren Raum einer Diskette mit Nullen	061
Terrain	Erzeugung von fraktalen Landschaften	061
ATPatch	Ändert IBM-Transformer für Workbench 1.2	061
Hack 1.03D	Das legendäre Abenteuerspiel	062
Larn 12.0b	Umfangreiches Abenteuerspiel mit Grafik	063
IFF	Diskette mit Informationen über IFF	064
Bawk	Textumformer, ähnlich UNIX-awk	065
Prep	FORTRAN-Vorübersetzer, Version 2.1	065
CloseWB	Schließen der Standard Workbench	065
MenuBuilder	Erstellt C-Strukturen für Menüs automatisch	065
MWB	Erstellen eines neuen Workbench-Schirmes	065
SunMouse	Mauszeiger verhält sich wie bei SUN üblich	065
PascalToC	Konvertiert Pascal nach C	065
JTime	Anweisungen zum Bau einer Echtzeit-Uhr	065
NewPackets	Beschreibt neue 1.2 Amiga-DOS Strukturen	065
RunBack	Führt Programm im Hintergrund aus	065
Cookie	Zeigt jedesmal einen neuen, lustigen Spruch	065
Melt	Gag: Läßt WB-Screen wie Wachs zerfließen	066
Nart	Ein kleiner Scherz	066
TagBBS 1.02	Ein Mailbox-System	066
Flip	Gag: Workbench steht auf dem Kopf	066
Dk	Gag: Bildschirm zerschneit	066
Assigned	Zeigt, ob Name schon zugeordnet wurde	066
SnapShot	Druckausgabe des Bildschirminhaltes	066
SendPackets	Erlaubt Senden von AmigaDOS-Packets	066
Asm68k 1.0.1	Makro-Assembler	066
Purty	Erlaubt bequeme Kontrolle eines Druckers	066
Free	Zeigt freien Platz auf AmigaDos-Gerät	066
AmScsi	Beschreibt Bau eines SCSI-Controllers	066
Foogol	Übersetzt foogol nach Vax-Assembler	066
MallocTest	Erlaubt Testen der Speicherverwaltung	066
RayTracer	Einfacher Ray-Tracer	066
AmigaSpell 2.0	Englischer Rechtschreibprüfer	067
Dux5	Erlaubt CLI-Dateioperationen über die Maus	067
Mandala	Erzeugt Mandala-Grafik mit Ton	067
PersMait	Demo einer Dateiverwaltung	067
AmCat	Programm zur Disketten-Katalogisierung	067

Wheel	Glücksrad-Spiel	067
Icons	Eine Sammlung von Icons	067
RSL-Clock 1.3	Uhr mit Zusatzfunktionen in der Menüleiste	067
HexCalc	Rechnen in verschiedenen Zahlensystemen	067
Comm 1.33	Terminalprogramm	067
RTCubes	Zeigt 16 Würfel in drei Dimensionen	067
Bouncer	Grafikdemo: Ball hüpft dreidimensional	067
MicroGNUEmacs	Amiga Version (MG) des bekannten Editors	068
Wc	Zählt Wörter und Zeichen in ASCII-Text	069
Sb	Erlaubt Betrachten von System-Strukturen	069
IconType	Ändert den Typ eines Icons	069
Conman 0.9	Erweitert das CLI zu einer Shell	069
Make	Make, für Manx, mit neuen Möglichkeiten	069
Spew	Erzeugt englische Zeitungsüberschriften	069
MouseClock	Maus-Zeiger wird zur Digital-Uhr	069
Console	Verbesserungen für Konsole	069
Asm68k 1.0.3	Makro-Assembler	069
Frags	Zeigt freie Speicherbereiche an	069
Dk	Gag: Bildschirm zerschneit	069
Spool	Drucker-Spooler-Programm	069
MonProc	Zeigt Prozeß-Paket-Austausch	069
BlitLab	Erlaubt Experimentieren mit dem Blitter	069
Lens 1.0	Vergrößert einen Bildschirm-Ausschnitt	070
SetKey 1.0	Demoversion eines Programms zum Tastenbelegen	070
GravityWars	Weltraum-Schlacht	070
DoTil	Erlaubt CLI-Dateioperationen über die Maus	070
Jobs	Erlaubt CLI-Dateioperationen über die Maus	070
Arc	Komprimierendes Dateiarchivierungsprogramm	070
BlackBook	Kleines Notizbuchprogramm	070
Logo	Interpreter für die Programmiersprache Logo	070
Life-3d	Dreidimensionale Simulation von Zellwachstum	070
AmigaMonitor	Zeigt Systemzustand dynamisch an	070
VPG 1.0	Testbild-Generator	070
NewFonts	Fonts, auch mit elektronischen Symbolen	071
AmiBas	Verschiedene AmigaBASIC-Programme	071
Blocks	Demo-Programm, ähnlich Lines	071
Comm 1.34	Terminalprogramm	071
TimeSet	Komfortables Stellen von Uhrzeit und Datum	071
PWDemo	PowerWindows, Demoversion 1.2	071
FPic	Zum Verändern bestehender IFF-Bilder	071
Icons	verschiedene Icons	071
PetCLI	Erlaubt Editieren im CLI, AmigaBasic	071
IconMk	Erzeugt Icons für Programme	071
DiskX	Brauchbarer Diskettenmonitor	071
Rot 0.5	Editieren und Anzeigen von 3D-Objekten	071

© 1990 technicSupport

AirFoil	Strömungsanalyse von Tragflächenprofilen	071
Pictures	Verschiedene Bilder	072
Add	Anwählen von Menüpunkten mit der Tastatur	073
PenPalFont	Zeichensatz, ähnlich einer Handschrift	073
Lmv	Zeigt mehrere IFF-Bilder als Film an	073
Dissolve	Weiches Einblenden von ILBM-Bildern	073
XPlor	Zeigt Systemlisten mit Adressen an	073
ParOut	Zeigt Benutzung der Parallel-Schnittstelle	073
AutoIconOpen	Öffnen von Icons ohne Maus	073
MouseOff	Der Mauszeiger wird nur bei Bedarf gezeigt	073
Expose	Alle wichtigen Fensterteile werden freigelegt	073
DTerm	Kleines, flexibles Terminalprogramm	073
TypeAndTell	Zeigt Installation eines Input-Handlers	073
RunBackGround	Führt Programm im Hintergrund aus	073
SnapShot	Druckausgabe des Bildschirminhaltes	073
Dio	Erlaubt leichten Zugriff auf EXEC-Geräte	073
Lit	Ausgabe einer binären Datei	073
Cled	Befehlszeilen-Editor	074
Random	Zufallszahlen-Erzeuger in C	074
Control	Erlaubt genauere Kontrolle über Druckausgabe	074
Funds	Hilfe zum Verwalten von Aktienkursen	074
DropShadow	Workbench Fenster erhalten einen Schatten	074
MCad 1.2.4	Sehr brauchbares CAD-Zeichenprogramm	074
TDebug	Dynamisches Betrachten eines Device	074
Less	Bequemes Betrachten von Textdateien	074
Units	Umwandlungsprogramm für Einheiten	074
Dme 1.25	Konfigurierbarer Texteditor	074
Makemake	Erstellt automatisch eine Make-Datei	074
XCopy	Kopieren mit Erhalt des Datums	074
Eless	Schnelle Verzeichnisausgabe, nicht für FFS	075
Copy	Kopierprogramm, erhält Datei-Datum	075
SpeedDir	Schnelle Variante des dir-Befehls	075
BSplines	Demonstriert B-Spline-Kurven	075
MouseOff	Mauszeiger wird nur bei Bedarf sichtbar	075
SetFont 2.0	Setzen eines neuen CLI-Zeichensatzes	075
Bezier	Demonstriert Bezier-Kurven	075
Comm 1.34	Terminalprogramm	075
Diff	Vergleicht zwei Dateien miteinander	075
Fd	Schnelle Verzeichnisausgabe	075
DuM2	Erlaubt CLI-Dateioperationen über die Maus	075
HardCopy	Protokolliert CLI-Sitzung auf dem Drucker	075
Draco1	Diskette 1 für die Sprache Draco	076
Draco2	Diskette 2 für die Sprache Draco	077
MandelVroom	Mandelbrot-Programm	078
Cycles 1.0	Ein aufregendes Reaktionsspiel	078

EOMS	Recht schwieriges Spiel, gute Grafik	078
Echo	Neue Version des CLI-Befehls, in Assembler	079
Info	Wie der CLI-Befehl, aber mit Quellcode	079
SetLace	Schaltet Interlace auf WB ein und aus	079
Kill	Entfernt einen Task so gut, wie möglich	079
MonProc	Verbesserte Version, zeigt Prozess-Pakete	079
Who	Zeigt Tasks an	079
M2Error	Zeigt Fehler des TDI Modula2-Compilers an	079
Uw 1.00	Unix Fenster client für Amiga	079
Why	Erklärt Fehlercodes	079
ScnSizer	Ändert Window-Maximalwerte in preferences	079
Cmd	Dirigiert Daten von Gerät nach Datei um	079
AssignDev	Ordnet einem Gerät verschiedene Namen zu	079
ParTask	Hilft beim Finden des erzeugenden Tasks	079
QueryAny	Erfragt Yes/No im CLI-Script	079
SharedLib	Beispiel einer teilbaren Bücherei	079
Mounted	Testet, ob ein Gerät im mounted-Zustand ist	079
Mounted	Testet, ob ein Gerät im mounted-Zustand ist	079
Nro	Formatierer mit Quellcode, ähnlich roff	079
AuxHandler	Erlaubt Anschluß eines externen Terminals	079
Task	Beispiel für die Funktion CreateTask()	079
ShowPrint	IFF-Bilder können damit gedruckt werden	080
PrtDrvGen	Erstellung beliebiger Druckertreiber	080
Othello	Othello Spiel, ohne Quellcode	080
NewDemos	Neue Demos, mit weniger CPU-Belastung	080
Conman	Erweitert das CLI zu einer Shell	080
Timer 1.5	Uhrprogramm	080
Sizzlers 1.7.0	Einige grafische Demos	080
CardFile	Handliche Dateiverwaltung, wie Karteikasten	080
PrinText	Zeigt ASCII-Datei auf Bildschirm an	080
IMandelVroom	Zeigt auch Innenraum der Mandelbrot-Menge	080
AmiGazer	Zeigt nächtlichen Himmel, kennt 1573 Sterne	080
Tools	Verschiedene nützliche Programme	080
RainBench	Workbench erhält Regenbogen-Hintergrund	080
ShortCut	Erlaubt Tastaturmakros im CLI	080
Smush	Gagprogramm	081
Scat	Gag: Windows flüchten vor Mauspointer	081
Target	Gag: Mauszeiger wird zum Fadenkreuz	081
Brushes	53 Pinsel für Dpaint, auch Schaltelemente	081
KeyLock	Sichert Amiga mit Paßwort	081
CheckIFF	Prüft IFF-Datei	081
Conman	Erweitert das CLI zu einer Shell	081
Fonts	Verschiedene Zeichensätze	081
Cled	Befehlszeilen-Editor	081
Icon 6.0	Programmiersprache Icon	081

© 1990 technicSupport

Asm68k 1.1.0	Makro-Assembler	081
AutoFacc	Schaltet FACC-Fenster autom. in Hintergrund	081
Sand	Gag: Mauszeiger ist magnetisch	082
Icons	Verschiedene neue Icons	082
Rocket	Gag: Rakete muß auf Fenster landen	082
Adventure	Das klassische Abenteuer-Spiel	082
D2D-Demo	Demo Version, Dateibehandlung	082
AmicTerm	Terminalprogramm	082
DiskMan	Erlaubt CLI-Dateioperationen über die Maus	082
Panl	Erlaubt Kontrolle über Midi-Geräte	082
DX-Synth	Stimmen-Speicher für Yamaha-DX-Synthesizer	082
TeX	TeX - Demo des Computersatzsystemes	083
PopCLI	Öffnen eines CLI-Fensters auf Tastendruck	084
BlitLab	Erlaubt Experimentieren mit dem Blitter	084
ViaCom	Gag: verschneiter Bildschirm	084
PipeHandler	Pipe-Gerät für Programmierer	084
Requester	Datei-Abfrager	084
HunkPad	Ändert die Hunks einer Ladedatei	084
GravityWars	Grafisches Geschicklichkeitsspiel	084
Ed	Einfacher, aber klassischer Editor	084
ScottDevice	SCSI-Treiber, zum Harddisk-Anschluß	084
AudioTools	Zeigt Zugriff auf Audio-Gerät	084
RawIO	Änderung der Standard-Eingabe nach CBREAK	085
LowMem	Informiert Programme bei knappem Speicher	085
Csh 2.06	Verbessert das CLI zu einer Shell	085
FileReq	Datei-Abfrager	085
Rocket	Gag: Rakete muß auf Fenster landen	085
Vnews	Zeigt Neuigkeits-Textdateien an	085
VMore 1.00	Zeigt Textdateien an	085
Plot6	Zeigt Sternenhimmel an	085
Hide	Für Programme, die nur mit max. 512 KB laufen	085
ImageTools	Intelligente Bearbeitung von IFF-Dateien	085
Cmd	Dirigiert Daten von Gerät nach Datei um	086
FileIISG-Demo	Demo einer Datenbank mit Bild und Ton	086
ClickToFront	Holt Fenster durch Doppelklick nach vorne	086
AutoPoint	Fenster unter Zeiger wird aktiviert	086
AdvSys	Programmiersprache speziell für Adventures	087
Tek4695	Ein Tektronix 4695/4696 Druckertreiber	087
Claz	Wandelt IFF-Bilder in Postscript-Files	087
Sc	Erzeugt Zufalls-Landschaften	087
Yaiffr	Liest IFF Bilder, auch HIRES, HAM und Overscan	087
Dme 1.27	Konfigurierbarer Texteditor	087
Zoo 1.42A	Komprimierendes Dateiarchivierungsprogramm	087
Commodities	Eine Amiga-Exec-Bücherei	087
Install	Zwei Ersatz-Programme für das Standard-Install	087

Diff	Vergleicht zwei Dateien miteinander	087
DropShadow	Workbench Fenster erhalten einen Schatten	087
WarpText	Programm für schnelle Textausgabe	087
ID-Handler	Erzeugt eindeutige Namen, AmigaDOS-Handler	087
MemWatch	Warnt Programme, wenn der Speicher knapp wird	087
MoveWindow	Bewegt Fenster ohne Mausbenutzung	087
AutoIconOpen	Kann Maus-Eingabe teilweise simulieren	087
WBDualPF	Demonstriert Dual-Playfield-Modus	087
MovePointer	Bewegt Zeiger ohne Mausbenutzung	087
PalTest	Testet, ob ein Amiga eine PAL-Maschine ist	087
MunchingSq	Gagprogramm	087
Elib	Beispiel für Aufbau einer Library (Manx C)	087
Snake	Einfaches Grafikdemo	088
FuncKey	Erlaubt freie Belegung der Funktionstasten	088
Snap	Erzeugt aus WB-Ausschnitt ein Icon	088
DirMaster	Programm zur Disketten-Katalogisierung	088
MFF-Demo	Demoversion der Dateiverwaltung Microfiche	088
ScreenShift	Erlaubt Adjustage des Bildschirmes	088
Snake	Einfaches Grafikdemo	089
AutoEnquirer	Schwächerer Requester	089
DirMaster	Programm zur Disketten-Katalogisierung	089
MFF-Demo	Demoversion der Dateiverwaltung Microfiche	089
ScreenShift	Erlaubt Adjustage des Bildschirmes	089
FuncKey	Erlaubt freie Belegung der Funktionstasten	089
DEMOlition	Gag: Kleiner Ball demoliert den Screen	089
PrinText	Bequemes Betrachten von Texten und Bildern	090
Timer 1.5	Uhrprogramm	090
CardFile	Handliche Dateiverwaltung, wie Karteikasten	090
AmiGazer	Zeigt nächtlichen Himmel, kennt 1573 Sterne	090
Sizzlers	Grafikdemo	090
Othello	Reversi-Spiel	090
SpaceAceDemo	Demo eines grafischen Action-Spieles	090
ShowPrint	IFF-Bilder können damit gedruckt werden	090
PrtDrvGen 2.2b	Erstellung beliebiger Druckertreiber	090
ShortCut	Erlaubt Tastaturmakros im CLI	090
Conman	Erweitert das CLI zu einer Shell	090
IMandelVroom	Leichte änderung des Mandelbrot-Generators	090
RainBench	Workbench erhält Regenbogen-Hintergrund	090
NewDemos	Neue Demos, mit weniger CPU-Belastung	090
ADL	Sprache zum Erstellen von Adventures	091
HunkPad	Ändert die Hunks einer Ladedatei	092
Bawk	Textumformer, ähnlich UNIX-awk	092
Parse	Parser für arithmetische Ausdrücke	092
SmallLib	Besonders kleiner Ersatz für amiga.lib	092
Less	Bequemes Betrachten von Textdateien	092

© 1990 technicSupport

As6502	Assembler für den 6502-Prozessor	092
Shar	Packen und Entpacken von Dateien	092
UUencode	Binäre Daten werden als Text dargestellt	092
Ndir	Implementiert 4BSDUNIX Verzeichnis Funktionen	092
MicroEmacs 3.8i	Editor, programmierbar	093
Dme 1.27	Konfigurierbarer Texteditor	093
AudioTools	Zeigt Zugriff auf Audio-Gerät	094
ClickUpFront	Holt Fenster durch Doppelklick nach vorne	094
HeliosMouse	Aktiviert Fenster, auf dem der Mauszeiger steht	094
Terrain3d	Erzeugt Landschaften	094
IFF2Ps	Wandelt IFF-Datei nach Postscript	094
ModulaTools	Verschiedene Hilfen für Modula-Programierung	094
MergeMem	Verschmilzt Einträge verschiedener Platinen	095
PrinterStealer	Lenkt Druckerausgabe in Datei um	095
Record-Replay	Aufzeichnen und Simulieren von Benutzereingaben	095
Journal	Aufzeichnen und Simulieren von Benutzereingaben	095
Cmd	Dirigiert Daten von Gerät nach Datei um	095
CygnusEdDemo	Demoversion (max. 5 KByte) eines guten Editors	095
GOMF 1.0	Verhindert Guru-Meditationen	095
AnimPlayer	Spielt "Animation"-Dateien ab	096
LineDrawer	Unterstützt Zeichnen einer Linie	096
HackBench	Neugeschriebene Workbench	096
WarpText	Programm für schnelle Textausgabe	096
PopUpMenu	Zeigt wie man Pop-Up Menüs implementiert	096
Label	Druckt Etiketten mit beliebigem Text	096
Tek4695	Ein Tektronix 4695/4696 Druckertreiber	096
TimeRam	Testet Geschwindigkeit des Speichers	096
Chess	Schach-Programm, ohne besondere Grafik	096
GraphIt	Plottet einfache Funktionen	097
Juggler	Jongleur-Animation, mit Sculpt 3D erstellt	097
Shm	Grafisches Demoprogramm, zeichnet Farb-Kurven	097
Splines	Zeigt Spline Kurven und Pop-Up-Menüs	097
CutAndPaste	Ausschneiden von Teilen einer Textdatei	097
MouseReader	Lesen von Textdateien mit der Maus	097
QBase	Verwaltungshilfe für Mail-Basen	098
HdDriver	Treiber für WD-1002-05 Hard Disk Controller	098
Backup	Sichert Inhalt einer Festplatte auf Disketten	098
DCDemo	Demoversion von DiskCat 2.3 Katalogisierer	098
Access! 1.34	Vielfarbiges ANSI-Terminalprogramm	098
Thai	Quizprogramm für Thai	098
A-Render	Ein Ray-Tracing Konstruktions-Werkzeug	099
Conman	Erweitert das CLI zu einer Shell	100
WBLander	Einfaches Spiel im WB-Screen	100
Berserk	Gute grafische 3D-Demo für Amiga	100
CirPlane	Landschafts-Generator für Videoscape 3D	100

© 1990 technicSupport

PsIntrp	Postscript-Interpreter	101
IconAssembler	Erlaubt das Verbinden von Piktogrammen	101
StartUps	Ersetzt Standard C-Start-Routinen	101
MicroSPELL	Rechtschreibprüfer für Englisch	101
Midi	Unterstützungs-Software für MIDI	101
SiliCon	Kommandozeilen-Interpreter als CLI-Ersatz	102
Sectorama	Hilft bei zerstörten Disketten	102
Xicon	Erlaubt Aufruf von CLI-Scripts über Icons	102
Match_Stuff	Perfektes Pattern-Matching Programm	102
Dbug	Unterstützt Testen von C-Programmen	102
Sol	Solitär-Spiel	103
MMF-Update	Eine MMF Erweiterung zum Text-Einbinden	103
Cref	Erstellt Cross-Reference-Listen für C	103
Pack-It	Packt gesamte Diskette zu einer einzigen Datei	103
IntuiDOS	Erleichtert das Arbeiten mit Dateien im CLI	103
Calc	Programmierbarer UPN-Rechner	103
AvlTrees	Implementation von AVL-Bäumen	103
DosKwik	Beschleunigt Diskettenzugriff (bis zu 3fach)	103
AnalytiCalc	Rechenblatt mit höchster Leistung	104
Freud	Persönlichkeitstest nach Freud	105
Curves	Witzige Anwendung von LeastSquare	105
GravityWars	Gravitationsabgelenkte Schüsse sollen treffen	105
Replay	Führt mit Record aufgezeichnete Eingaben aus	105
L	Verzeichnis-Ausgabe	105
IPo2C	Wandelt Intuition-Zeiger nach C-Quellcode	105
SharpFont	Zeichensatz	105
FlamKey	Erlaubt es, den Amiga zu sperren	105
LeastSquare	Kurven nach Methode der kleinsten Quadrate	105
Pere-et-Fils	Zeigt das Arbeiten mit reentrantem Code	105
Record	Zeichnet alle Eingaben auf (s. Replay)	105
Clear	Löscht freien Speicherplatz	105
Stack	Ersatz für das Standardprogramm Stack	105
AsmFilter	Hilft bei Assembler Re-Assemblierungen	105
AddKickMem	Erklärt Kickstart-Speicher als unbelegt	105
DrunkenMouse	Gag: Mauszeiger torkelt	105
Bison	Ein Parser-Generator, wie UNIX-yacc	105
FuncKey	Erlaubt freie Belegung der Funktionstasten	106
RistiNolla	Das Spiel Gomoku	106
MoreArt	Verschiedene Bilder	106
QuickFlix	Für Animationen mit IFF-Bildern	106
SVTools	Verschiedene nützliche Programme	107
Diff	Vergleicht zwei Dateien miteinander	107
ProSuite	Programme und Informationen für Programmierer	107
Csh 2.07	Verbessert das CLI zu einer Shell	107
DirMaster	Programm zur Disketten-Katalogisierung	108

© 1990 technicSupport

Zoo	Komprimierendes Dateiarchivierungsprogramm	108
PrintPop	Schickt Steuersequenzen an den Drucker	108
Alist	Gibt den Inhalt von Verzeichnissen aus	108
Dots-Perfect	Drucker-Treiber für Epson MX80	108
Tek	Tectronics 4014 Terminalemulator	108
Sectorama	Diskettenmonitor mit Zugriff auf Festplatten	108
MonIDCMP	Zeigt IDC-Nachrichten eines Fenster IDCMP	108
UUCp	Erlaubt Benutzung des Amigas als Netz-Knoten	109
SimCPM	Ein CP/M Simulator (8080 mit H19 Terminal)	109
Machine	Sehenswerte Animation für 1 MB	109
A68k	Amiga-verträglicher Assembler	110
Pdc	Ein C-übersetzer für den A68k und Blink	110
AssignDev	Ordnet einem Gerät verschiedene Namen zu	111
Labels	Erstellt Kreuzverweis von Systemkonstanten	111
Amyload	Zeigt Amiga Belastung an	111
Mandel	Erzeugt Abbildung der Mandelbrotmenge	111
HeliosMouse	Aktiviert Fenster, auf dem der Mauszeiger steht	111
Gauge	Zeigt Speicherbedarf grafisch an	111
PopLife	Ein Game Of Life startet auf Knopfdruck	111
ViaCom	Gag: verschneiter Bildschirm	112
RGB	Ein Amiga-Computerfilm, sehr hübsch	112
WaveBench	Gag: Workbench erhält Welleneffekt	112
Focus	Ein Amiga-Computerfilm, sehr hübsch	112
Bully	Hilft beim Zeigen verschiedener Demos	112
DropShadow 2.0	Workbench Fenster erhalten einen Schatten	112
BeachBirds	Ein hübscher kleiner Gag	112
M2Amiga	Ein Modula-Compiler (nur für kleine Programme)	113
AmiCron	Startet Programme zu bestimmten Zeitpunkten	113
NoIconPos	Löscht Piktogramm-Positionen	113
Dme 1.28f	Konfigurierbarer Texteditor	113
DosDev	Beispiel für einen DOS-Treiber	113
WBLander	Einfaches Spiel im WB-Screen	114
CDecl	Hilft beim Erlernen von C-Deklarationen	114
VT100 2.7	Terminalprogramm, VT100-Emulation	114
Marketroid	Böse Animation über Amiga-Marketing	115
Killer	Eine sehr schöne Demo	115
Movies	Ein Animations Abspieler mit Beispielen	116
AMUC_Demo	Zeigt ein sehr langes Bild	117
ExP_Demo	Vorführversion eines Malprogrammes	117
HAMmmm	Eine Demo mit Klang und Grafik	118
WireDemo	Zeigt sehr schnelle 3D-Animation	118
Stars	Sterne fliegen auf den Betrachter zu	118
Empire	Sehr umfangreiches Strategiespiel in Draco	118
MicroEmacs 3.9	Editor mit umfangreicher Dokumentation	119
Bankn	Verwalten eines (Giro)Kontos	120

© 1990 technicSupport

BackGammon	Das Brettspiel Backgammon	120
EgyptianRun	Spiel, bei dem man durch eine Wüste fährt	120
Amoeba	Sehr guter SpaceInvaders-Clone	120
IconImage	Ändert Piktogramm-Bild für Workbench	120
Plot	Plottet dreidimensionale Funktionen	121
Uedit 2.3	Flexibler ASCII-Texteditor	121
BasicStrip	Konvertiert andere Dialekte nach AmigaBasic	121
WBColors	Programmen wird Farbeinstellung zugeordnet	121
Stairs	Erzeugt Ton-Illusion	121
DataPlot	Plottet Daten aus	121
IFF2Pcs	Erzeugt aus einem IFF-Bild ein Puzzle	122
Asteroids	Spiel, ähnlich Cosmoroids, mit Sound	122
PushOver	Strategiespiel in AmigaBasic	122
Names	Erstellt Versandlisten	122
PuzzlePro	Generiert ein Puzzle aus IFF-Bildern	122
Pr	Ausdrucken von Textdateien	122
Car	Ein unterhaltsamer Computerfilm	123
Arp	Neugeschriebene Standard Amiga-Befehle	123
Icons	Verschiedene Piktogramme für die Workbench	124
Tarot	Spiel in AmigaBasic	124
ElGato	Animation einer laufenden Katze (Videoscape 3D)	125
Colour	Ändert die Farben eines Screens	126
Suplib	Bücherei mit verschiedenen Funktionen	126
Iconify	Erstellt ein Workbench-Piktogramm	126
HBHill	Halfbrite-Animationsdemo	126
Dance	Demo mit verschiedenen tanzenden Polygonen	126
VCheck	Virusdetektor von Commodore, Version 1.2	126
OnlyAmiga	Bälle werden von Pyramiden jongliert	126
Ripples	Computerfilm	127
Nemesis	60 Sekunden-Weltraum-Computerfilm mit Sound	127
Bounce	Graphische Demo	127
SEd	Sehr einfacher Texteditor	128
MRBackUp 1.3	Zum Erstellen von Festplatten-Sicherungskopien	128
LedClock	Einfache Zeitanzeige	128
WKeys	Erleichtert Umgang mit Screens und Windows	128
DirMaster 1.1	Programm zur Disketten-Katalogisierung	128
Dis	Speicherdisassembler für 68000er-Code	128
DropCloth 2.2	Unterlegt Workbench mit IFF-Bild	128
PrtDriver	Druckertreiber für Toshiba-Drucker	128
SimplePaint	Sehr einfaches Zeichenprogramm	128
SdBackUp 1.1	Zum Erstellen von Festplatten-Sicherungskopien	128
PaintJet	Druckertreiber für den HP-PaintJet-Printer	129
DosKwik	Beschleunigt Diskettenzugriff (bis zu 3fach)	129
Patch 2.0	Zum Updaten von Dateien mit Diff-Files	129
MRBackUp 2.1	Zum Erstellen von Festplatten-Sicherungskopien	129

© 1990 technicSupport

MRBackUp 2.0	Zum Erstellen von Festplatten-Sicherungskopien	129
Patch 1.3	Zum Updaten von Dateien mit Diff-Files	129
Mach 1.6a	Zur Mausunterstützung, mit Hotkey-Funktionen	130
Evo	Zeigt die Entwicklungsformen des Menschen	130
HP 1.0	Wissenschaftlicher Taschenrechner	130
MachClk 1.2a	Zur Mausunterstützung, mit Hotkey-Funktionen	130
PatEdit	Zum Erstellen von Patterns (Füllmustern)	130
QMan	Schnelles Mandelbrot-Programm	130
WFrags	Aktuelle Anzeige über Speicherausnutzung	131
MicroGNUEmacs1b	Amiga Version (MG) des bekannten Editors	131
DFC	Kopierprogramm als Ersatz für WB-DiskCopy	131
Life	Life-Simulation	131
Mackie	Startet ein CLI über Tastenkombination	131
HyperBase 1.6	Interessante Dateiverwaltung	131
Showanim 4.00	Vorführprogramm für Videoscape 3D-Files	132
Berserk	Sehenswerte Videoscape 3D-Animation	132
Overscan	Patcht das System von 200 auf 256 Zeilen	133
CRCLists	Checksummenliste für FishDisks 1 bis 128	133
CRC	Bildet Checksumme über ganze Diskette	133
ConMan 1.1	Erweitert das CLI zu einer Shell, sehr gut	133
Movie 1.01	Zum Abspielen von Dilbm/Pilbm-Animationen	134
SmartIcon 1.0	Windows werden platzsparend als Icons abgelegt	134
Find	Sucht in allen Verzeichnisse nach best. File	134
Dme 1.29	Konfigurierbarer Texteditor	134
Browser 1.2	Programmaufruf über Pull-Down-Menü	134
BoingThrows	Sculpt/Animate 3D-Animation	134
TeXF 2.5	Konvertiert AmigaTeX-Fonts, mit vielen Fonts	135
TheLibrary 2.8	Speichert verschiedene Dateiformate einheitlich	135
Paste	Verändert das Aussehen einer Textdatei	136
Less 1.2	Bequemes Betrachten von Textdateien	136
AsmToolBox	Verschiedene Assembler-Quellcodes	136
Bison	Ein Parser-Generator, wie UNIX-yacc	136
YaBoing II	Gag: sich bewegende Zahlen erhaschen	136
Zoo 1.71	Komprimierendes Dateiarchivierungsprogramm	136
IFF2Pcs 1.1	Erzeugt aus einem IFF-Bild ein Puzzle	136
VirusX 1.0	Sucht nach Viren im Speicher und auf Disk	137
VLabel 1.20	Dient der Erstellung von Diskettenaufklebern	137
VGad 1.0	Zum Erzeugen von Gadget-C-Quellcodes	137
Muncho	Gag: Beim Wechseln einer Disk ertönt Sound	137
CT	Image-Prozessor, zum Ansehen von CT-Dateien	137
JeansIcons	Viele neue Icons	137
Sit 1.10	Ändert den Typ eines Icons	137
ModulaTools	Verschiedene Modula 2-Routinen zum Einbinden	138
MacFont	Konvertiert Mac-Zeichensätze ins Amiga-Format	138
VT100 2.8	Terminalprogramm, VT100-Emulation	138

Diff	Vergleicht zwei Dateien miteinander	138
ForEach	Zum Komprimieren von Dateien	138
VT100 2.6	Terminalprogramm, VT100-Emulation	138
AmigaLine	Texte, was Programmierer beachten müssen	138
RemLib	Entfernt Bibliotheken aus dem Speicher	139
WheelChairSim	3D-Grafikdemo, Bewegung in Gängen	139
AmiCron 2.5	Startet Programme zu bestimmten Zeitpunkten	139
ProCalc 1.0	Programmierbarer Taschenrechner	139
WArranger	Bringt Fenster nach vorn oder hinten	139
ListScanner	Gibt komplette Exec-Liste mit Adressen aus	139
TurboBackup 1.00	Kopierprogramm für maximal 4 Laufwerke	139
SBProlog 2.3.2	Vollständiger Prolog-Compiler	140
Small-C	Einfacher C-Compiler für kleine Programme	141
SBProlog-Src	Quelldateien zum SB-Prolog-Compiler	141
Diff	Vergleicht zwei Dateien auf Gleichheit	142
FracGen 1.1	Erzeugt fraktale Gebilde nach beliebigen Formeln	142
SciSubr	Verschiedene Fortran-Programmodule	142
RIM	Umfangreiche relationale Datenbank	143
AnalytiCalc	Multiplanähnliche Tabellenkalkulation	144
DMouse 1.06	Mausunterstützung mit PopCli, Hotkeys und mehr	145
DNet 1.2	Für Datenaustausch mit UNIX-Systemen	145
VT-Prolog	Vollständiger Prolog-Compiler	145
Tab	Zum Experimentieren mit Gitarren-Sounds	145
Csh 2.07m	Cli wird zu einer Shell	145
CRCLists	Checksummenliste für Fishdisks 129 bis 146	146
C-Light 1.0	Demoversion eines Ray-Tracing-Programmes	146
MemoPad 1.1	Elektronischer Terminkalender	146
DmeMacros	Erweitert Matt Dillons' Editor Dme	146
Blanker2 1.27	Schaltet den Bildschirm nach 90 Sekunden dunkel	146
MG2b	Neue Version des MicroGNUEmacs-Editors	147
FME	Landschaftseditor für das Spiel FirePower	148
Scrambler 0.01	Verschlüsselt beliebige Dateien	148
HandyIcons	Erlaubt Programmaufrufe von der WB-Menüzeile	148
Escape from Jovi	Geschicklichkeitsspiel mit Schwerkraftsimulation	148
SSP	Abspielprogramm für digitalisierte Sounds	149
AnimalSounds	Digitalisierte Tiergeräusche	149
Scheme 1.2	Demoversion eines Elektronik-CAD-Programms	149
DX-VoiceSorter	Sortierprogramm als Zusatz zum VoiceFiler	149
Keep 1.2	Hilfsprogramm für DFÜ-Downloads	149
Less 1.3	Bequemes Betrachten von Textdateien	149
Pilot	Demo der Programmiersprache mit Demoprogrammen	150
Iconizer	Aufbau einer Mauspointer-Sammlung	150
ExecLib	Beispiel, wie Libraries programmiert werden	150
AirFoil	Strömungsanalyse von Tragflächenprofilen	150
DC10	Brauchbarer AmigaBasic-Flugsimulator	150

© 1990 technicSupport

StealMemBoot	FastMem wird von Bootblockprogramm entfernt	150
Icons	Viele neue Icons	151
SCT	Einfaches Neudefinieren von Screenfarben	151
PCopy 1.0	Nützliches Kopierprogramm mit Extras	151
Surveyor	Zeigt Mauskoordinaten auf jedem Screen an	151
SlideShow	Gemischte Bilder aus allen Bereichen	151
Globe	Grafikdemo einer rotierenden Weltkugel	151
RunBack	Führt Programm im Hintergrund aus	152
Blk	Erstellt einfach Requester-C-Quellcodes	152
UUCp	Erlaubt Benutzung des Amiga als Netz-Knoten	152
HP11C	Emuliert einen HP11C-Taschenrechner	153
HPMan 1.1	Beeinflussen von Laserdrucker-Parametern	153
Synthemania	Mit Soundtracker erstelltes Musikstück	153
Dme 1.30	Konfigurierbarer Texteditor	153
Guardian 1.1	Sehr guter Virus-Schutz für Kick 1.2	154
DiskLib	Druckt kompletten Verzeichnisbaum aus	154
DClock	Einfache Uhr für die Menüzeile	154
UnDelete	Restauriert gelöschte Dateien	154
ADA	Ein ADA-Syntax-Checker	154
VirusX 1.21	Sucht nach Viren im Speicher und auf Disk	154
Print_Spool	Drucken von Textdateien als Spooler-Task	154
VirusAlert 1.01	Virusschutz durch signifikanten Bootblock	154
Vs 2.01	Sucht nach Viren im Speicher und auf Disk	154
Cal	Gibt Monatskalender aus	154
WhereIs	Sucht auf Geräten nach einer Datei	154
WIcon 1.14	Bringt alle WB-Fenster auf Icongröße	154
AssemblyDemos	Verschiedene Assembler-Demoprogramme	154
Bison	2 neue Versionen des Parser-Generators	155
AsmExamples	Beispiele einfacher Assembler-Codes	155
NoSmoking	Beispiel für recoverable Alert	155
Scenery 1.0	Erzeugt gelungene Zufallslandschaften	155
Go64	Spaßige Emulation eines C64	156
Grammars	Ada-, C-, und Pascal-Files für Flex und Bison	156
oops!	Gag: Hinterlegt WB mit abgestuften Farben	156
Blocks2	Kleine graphische Animation	156
Flex	Scanner-Generator, wie UNIX-lex	156
ScreenZap	Entfernt alle Screens hinter der Workbench	157
60or80	Wahl der Spaltenanzahl ohne Preferences	157
BootBack	Kopiert Bootblock in Datei und zurück	157
AmicForm	Erstellt Phonebook für AmicTerm	157
SetPrefs	Lädt neue system-configuration-Datei	157
AnimBalls	Animation mit Benutzerinteraktion	157
ECPM	Emuliert 8080 mit H19-Terminal	157
KeyFiler	Sortiert BBS-Nachrichten	157
Xicon 2.01	Erlaubt Aufruf von CLI-Scripts über Icons	157

© 1990 technicSupport

ScreenX 2.1	Manipulieren, Saven und Drucken von Screens	158
PCBTools	Demo eines Platinenlayout-Programms	158
TaskX 2.0	Anzeigen und Manipulieren von Prioritäten	158
MemBoardTest 2.4	Testet A1000-Speichererweiterungen	158
VirusX 1.6	Virusdetektor für die Workbench	158
YachtC3	Würfelspiel mit scheußlichem Sound	158
DiskX 2.1	Verbesserte Version des Disk-Editors	158
MSDOS 0.1	Liest und konvertiert MSDOS- und ST-Formate	158
Free	Zeigt freien Speicher und Taskanzahl an	159
StarChart 1.2	Zeigt und identifiziert etwa 600 Sterne	159
TaskControl	Erlaubt Kontrolle über Tasks	159
TUC	Uhr mit Speicherplatzanzeige	159
MidiTools	Verschiedenes für MIDI-Anwendungen	159
Calls	Gibt Struktur von C-Sources aus	160
Dis	Speicherdisassembler für 68000er-Code	160
M4	Makroprozessor, wie UNIX-M4	160
MemoPad 1.2	Elektronischer Terminkalender	160
DMouse 1.09	Mausunterstützung mit PopCli, Hotkeys und mehr	160
NeuralNets	Simuliert Modell des menschl. Lernverhaltens	160
DWIP	Druckt IFF-Bilder auf Typenraddrucker	160
Check 1.03	Syntaxprüfung von C-Sourcecodes	160
IncRev	Erhöht beim Kompilieren die Versionsnummer	161
Friends	Gag: Mauszeiger bekommt Gesellschaft	161
Perl	Zur Transformation von Textdateien	161
Mackie 1.1	PopCLI-Variante mit zusätzlichen Funktionen	161
LGZ	Erzeugt Karten für das LGZ-Spiel	161
Nag 1.6	Kalenderprogramm, Eintragungen möglich	161
VBoot	Macht den Bootblock zu einem Executable	161
GetSprite	Konvertiert DPaint-Brushes in C-Source	161
VRTest	Virusprotektor	161
SetPrompt	Setzt Prompt in anderer Schrift und Farbe	162
UnknownGirl	Spielt kurzes Musikstück	162
Install	Bootblock schaltet FastMem und ext. Floppies ab	162
Split	Trennt eine Datei in zwei Teile	162
Ins_Returns	Fügt Returns in Dateien ein	162
TES 1.0	Gadgets für Titelleiste von CLI-Fenstern	162
PWFix	Beseitigt Fehler in PowerWindows(1.3)-Sources	162
Avi 1.0	Amiga-Implementation des UNIX-vi-Editors	162
IFFar	Archiviert IFF-Dateien	162
SetPALorNTSC	Zum Testen von Programmen unter PAL und NTSC	162
LabyrintII	Textadventure	162
Tree	Gibt Verzeichnishierarchie eines Gerätes aus	162
Flow2Troff 1.0	Macht Flow-Dateien zu Unix-Troff-Files	162
Filter	Entfernt alle Nicht-ASCII-Zeichen aus Datei	162
XChar	Entfernt alle Nicht-ASCII-Zeichen aus Datei	162

© 1990 technicSupport

TDemo	Extrem schnelle Blitter-Demo, sehenswert	162
Makexe	Startet Batchdateien von WB und CLI	162
WordCount	Zählt Wörter und Zeichen in ASCII-Text	162
Dark	Kleine Animation	162
StripBin	Entfernt oder wandelt Nicht-ASCII-Zeichen	162
FiveInLine	Ristinolla-Spiel	163
View	Einfachster Textbetrachter	163
MachII 2.4c	Maus-Utility mit PopCli, ClickTOFront etc.	163
Bankn 1.5	Verwalten eines (Giro)Kontos	163
MemTrace	Kontrolliert Speicherbelegung und -freigabe	163
PcPatch	Patcht PCCopy/PcFormat auf 3 1/2-Zoll	163
ReadmeMaster	Stichwortverzeichnis für Contents-Dateien	163
PolyRoot	Berechnet Nullstellen von Polynomen	164
DiskSalv 1.3	Restauriert Disk und Festplatten (auch FFS!)	164
Zoo 2.00	Komprimierendes Dateiarchivierungsprogramm	164
NewZap 3.18	Dateiorientierter Diskettenmonitor	164
C-Functions	C-Funktionen für Lattice 3.10	164
PrtDrivers	Druckertreiber für DEC LN03+ und MT420d	164
Hed	Einfacher Texteditor	164
PcView 2.01	Betrachten von IFF-Bildern mit EGA-PC	164
Newton	Berechnet Nullstellen von Polynomen	164
Parsnag/Separate	Unterstützt Farbtrennung beim Epson JX-80	165
SPUDclock	Sagt Uhrzeit in bestimmten Abständen an	165
CPM	Emuliert CPM-Computer mit ADM3A-Terminal	165
RamCopy	Kopiert ganze Diskette über das Ram (ab 1 MB)	165
Conman 1.3	Erweitert das CLI zu einer Shell, sehr gut	165
PlotView	Zeichnet UNIX plot format-Dateien	165
Stevie 3.10a	Amiga-Version des UNIX-vi-Editors	166
MultiCalc	Emuliert Rechner mit RPN-Notation	166
Cref	Erstellt Cross-Reference-Listen für C	166
AutoGraf	Zeigt Bezinkostenverteilung über die Jahre	166
CloseMe	Gag: Fenster lassen sich nicht schließen	167
MRPrint	Kontrolliertes Ausdrucken von Dateien	167
Smus 3.6a	Zum Abspielen von IFF-Musikstücken	167
DSM	Erstellen von selbst ablauffähigen Sounds	167
CDecl	Übersetzt Englisch nach C	167
Sounddemos	Vier Musikbeispiele von Visionware	167
CLIcon	Startet CLI-Kommandos über Icons	167
MattDillon1	Alle Programme und Sources von M. D. (s. 169)	168
MattDillon2	Alle Programme und Sources von M. D. (s. 168)	169
Dis6502	Disassembler für 6502-Prozessor	170
MRBackUp 2.4	Zum Erstellen von Festplatten-Sicherungskopien	170
PtrAnim	Zum Erzeugen von animierten Mauszeigern	170
Surf	Erzeugt Bezier-Flächen	170
FastText	Assembler-Routine für schnelle Textausgabe	170

Turbo	Schaltet verschiedene DMA-Zugriffe ab	170
Aftterm	Emulation eines IBM 3278-Terminals	170
AZComm	Terminal-Programm mit ZModem-Protokoll	171
Sozobon-C	Atari-Freeware-C-Compiler	171
Xoper 1.2	Einfacher System-Monitor	171
Maze	Labyrinthe in 2D- und 3D-Darstellung	171
DataToObj	Konbertiert Raw-Data in Objectfiles (Manx)	172
PopInfo 2.0	Informiert über Laufwerke, Speicher und Zeit	172
MFix 1.00	Beschleunigt Marauder um bis zu 25 %	172
ProCalc 1.2	Programmierbarer Taschenrechner	172
Spiff	Für Dateivergleich, ähnlich UNIX-diff	172
HandShake 2.12a	VT52/100/102/220-Terminalemulation	172
Tetrix	Denk- und Reaktionsspiel (wie Tetris)	173
Uedit 2.4g	Texteditor, als Beta-Release	173
IFFLib	Library für Manipulationen an IFF-Dateien	173
FileSelect	Dateiauswahl-Requester in Assembler	173
ILBM2C	Wandelt IFF-ILBM-Datei in C-Code	173
CrcLists	Crc(s. Fish 133)-Checkliste für Disks 146-172	173
TitlePage	Druckt große Banner aus Einzelbuchstaben	174
Tunnel	Modula-2-Grafikdemo mit Source	174
Castle	Adventurespiel mit Grafik	174
IFF2Sun	Wandelt IFF-Bild in SUN-Rasterfile-Format	174
Print 1.2	Kontrolliertes Ausdrucken von Textdateien	174
Strings 1.0	Extrahiert Texte aus binären Dateien	174
Sh	Erleichtert das Entpacken von Archiven	174
Elements	Periodensystem mit vielen Stoffeigenschaften	175
Zippy 2.5	Ermöglicht CLI-Dateioperationen über die Maus	175
Furnish	Grafische Planungshilfe für Innenarchitekten	175
VirusX 3.10	Erkennt auch das IRQ-Virus	175
WBDepth	Ändert die Zahl der Workbench-Farben	175
Plot 4.1	Berechnet und plottet 3D-Funktionen	175
SaveBoot 2.2	Komfortables Archivieren von Bootblöcken	175
HyperNet	Zur Vernetzung mehrerer Tasks	176
AnalytiCalc	Tabellenkalkulationsprogramm	176
DiskSalv 1.32	Wie Version 1.3, von Fehlern befreit	177
ASpice	Simuliert elektronische Schaltkreise (ab 1.5 MB)	177
Path	Suchpfad wird über ein Device verwaltet	177
JAsk	Wie Ask, Antwort erfolgt jedoch über Requester	177
Marge	Fügt am linken Rand Leerzeichen in Datei ein	177
RexxArpLib 2.0	Library und Makros für ARexx	178
Cosmic	Weltraum-Strategiespiel	178
CIATimer	Mikrosekunden-Timer in C (Manx)	178
AmicForm 1.4	Erstellt Phonebook für AmicTerm	178
RemLib 1.11	Entfernt Library aus dem Speicher	178
Ls 2.0	Directory-Lister, 1.3-kompatibel	178

© 1990 technicSupport

BlackBox	BlackBox nach versteckten Atomen durchsuchen	178
Launch	Zum Starten von WB-Applikationen aus dem Cli	179
Regexp	Unix-regexp(3)-Funktionen für C	179
KickFont 3.0	Ersetzt Topaz-Font auf Kickstart-Diskette	179
Exception 0.6	C-Routinen zur Behandlung einiger Fehler	179
UnixUtil	Wc, Head, Tail, Detab, Entab, Trunc	179
DietAid 3.1	Hilft beim Einhalten einer Diät	179
TSnip 1.4a	Schneidet Texte aus beliebigem Fenster aus	179
Dmake	Beta-Version eines Make-Programms	179
GPrint 2.03	SW-Druckerhilsprogramm für Epson-kompatible	180
Jed 1.0	Menügesteuerter Editor, kann keine Umlaute	180
GeoTime 1.0	Zwei Weltzeituhren mit Grafik	180
TrekTrivia	Quizprogramm: 100 Fragen zu StarTrek	180
NoVirus 1.56	Umfangreiches Anti-Virus-Programm	180
RepString 1.0	Sucht und ersetzt Textteile in Datei	180
Browser 1.6	Dateinamenorientierte Workbench-Variante	180
Tracker 0.0a	Prüft C-Programme auf korrektes Verhalten	181
AMXLISP 2.00	Interpreter für die Sprache XLisp	181
Bally 0.1	Umsetzung des Arcade-Spiels Click	181
Iconmerger 2.0	Erstellt animierte Icons aus zwei Bildern	182
Edimap 1.0	Editor für Tastaturbelegungstabellen	182
SetFont 2.5	Aktiviert andere Zeichensätze auf der WB	182
HR 136	Täuscht in Malprogrammen mehr Farben vor	182
Sam 1.0	IFF-Klangspieler mit Beispielen	182
AMC 1.0	Scrollt Texte über farbigen Hintergrund	182
Mklib	Beispiel einer C-Library	183
FixFd	Wandelt Fd-Dateien in Assembler-Include-Files	183
PCQ	Wandelt Pascal-Code in Assembler (A68k)	183
BI 1.0	Wandelt IFF-Brushes in C-Code	184
CardMaker 1.0	Erzeugt USHORT-Datas für Spielkarten-Images	184
DPS 1.0	Bilder zeigen sich selbst an (ohne Hilfsprog.)	184
VacBench	Gag: Mauszeiger wird zum Staubsauger	184
Print 1.0	Ersetzt "copy to prt:"-Kommandozeile	184
World 1.02	Einfaches Textadventure	184
MouseUtil 1.1	Zum Ändern der Mausgeschwindigkeit	184
IFF-Info	Offizielle IFF-Disk von Commodore (Nov 1988)	185
Cards'O'Rama	Kartenspiel, der Guru spielt mit	186
Qt2	Sagt Uhrzeit in üblicher Manier an (englisch)	186
A68k 2.42	Metecomco-kompatibler Assembler	186
SimCPM 2.3	Emuliert 8080-Prozessor mit H19-Terminal	186
Mackie 1.13	PopCli-Variante mit zusätzlichen Features	187
DiskPerf	Benchmarktest für Laufwerke, auch mit FFS	187
SetCPU 1.4	Prüft auf andere CPU's und Koprozessoren	187
HackLite	Erweiterte Version des Hack-Adventures	187
Null 0.0	Ersatz für das NIL:-Gerät	188

© 1990 technicSupport

DiffDir 1.0	Vergleicht den Inhalt zweier Verzeichnisse	188
FastGro 1.0	Fractal-Generator	188
ExecDis 1.0	Disassembliert und kommentiert exec.library1.2	188
TextDisplay 1.1	Zeigt Texte an, mausgesteuert	188
BootIntro 1.0	Kreiert Vorspann mit kurzem Text	188
MemoryClock	Uhrzeit- und Speicherplatzanzeige	188
FracGen 1.23	Erzeugt fraktale Grafiken	188
MinRexx	Einfaches ARexx-Interface	188
NetHack 2.2	Erste Hälfte einer Hack-Version mit Sources	189
Mackie 1.2	Jetzt mit abwechslungsreicherem Screen-Blanker	189
Uedit 2.4g	Texteditor, ersetzt Beta-Release von Fish 173	189
NetHack 2.2	Zweite Hälfte einer Hack-Version mit Sources	190
GaryIcons	Einige mittelmäßige Icons	190
ILBM2Image	Wandelt IFF-Bild in C-Sourcecode	190
BlitLab 1.4	Erlaubt Experimentieren mit dem Blitter	191
ISpell 2.0.02	Englischer Rechtschreibprüfer	191
BCK	Erzeugt C-Codes für Requester	191
Pz15	Knobelspiel	191
FileBootBlock	Wandelt Bootblock in ausführbares Programm	191
Eval	Lattice-C-Routinen für mathematische Ausdrücke	192
ReSourceDemo	Demo eines interaktiven Disassemblers	192
PacMan87	Ein Pac-Man-Spiel	192
KeyMapEd 1.02	Editor für die devs:keymaps-Dateien	193
Zc	Sozobon C-Compiler, jetzt A68k-kompatibler Code	193
Moria 3.0	Dungeon-Adventure in der Art von Hack und Larn	194
MicroEmacs 3.10	Neue Version des bekannten Texteditors	195
HamPics	5 edle Ham-Bilder (von Sun-Workstation)	196
LoadPic	Zeigt Overscan-IFF-Bilder an	196
Stevie 3.35a	Amiga-Version des UNIX-vi-Editors	197
Nro 1.5	Formatierer mit C-Quellcode, ähnlich roff	197
CTags 4.7	Erzeugt tags-dateien aus versch. Sources	197
Find 1.2	Sucht in allen Verzeichnissen nach best. File	197
FixHunk	Programme laufen in erweitertem Speicher	197
Charon	Illustriertes Buch erzählt eine Geschichte	198
MIDIsoft 1.0	Überträgt Samples zw. Amiga und Roland S-220	199
SnipDemo 1.23	Graphische Darstellung verschiedenster Daten	199
Pyro 1.1	Bildschirmschoner mit Feuerwerk-Effekten	199
ASimplex 1.2	Simplex-Algorithmus für lineare Programme	199
Sho	Kurzer Bildanzeiger in Assembler	199
Csh 3.02a	Verbessert das CLI zu einer Shell	199
NotBoingAgain	Amiga-User löst Guru-Meditation aus	200
Tank	Amiga-Fisch harpuniert andere Fische	200
Draco 1.2	C-ähnliche, vielseitige Programmiersprache	201
DropCloth 2.4	Unterlegt Workbench mit IFF-Bild	201
SlavicFonts 1.0	Zeichensätze für slawische Sprachen	202

© 1990 technicSupport

Vlt 3.656	Terminalprogramm (VT 100) mit ARexx-Port	202
SmallLib	Kleiner Ersatz für die Amiga.lib	203
SpeechToy	Sprachausgabeprogramm mit Assembler-Source	203
ManxPCBug.asm	Report über einen Fehler im Manx-Compiler	203
AudioDev	Verbessertes Audio.device	203
TopTenLists	30 humorige Sprüche zum Thema Computer	203
GurusGuide	Alle Sources aus Gurus Guide, Meditation #1	203
FileIO	Dateiauswahlrequester in C und Assembler	203
YachtC3	Assembler-Variante des Spiels von Disk 158	203
TypeAndTell	Beispiel eines Input-Handlers in Assembler	203
Task	Assembler-Beispiel für CreateTask-Funktion	203
8SVX	Disassembliertes und kommentiertes 8SVX-Sample	203
Isam 0.9	C-Funktionen für indexseq. rel. Datenbanken	203
Samp	Ersatzformat für 8SVX-Samples	203
HAMCu	Workbench erhält 4096-farbigen Hintergrund	204
JPClock	Uhr mit Zusatzfunktion für die Menüleiste	204
FileReq	Dateiauswahlrequester mit Source in C	204
MouseBounce	Gag: Jagt Mauszeiger hin und her	204
PopInfo	Info-Befehl für WB, leider mit Guru	204
Teacher	Gag: Simuliert Guru-Meditation und Reset	204
PopDir	Dir-Kommando und Linkvirus-Check für die WB	204
GnuGrep 1.3	Grep-Kommando des GNU-Projectes	204
Image-Ed 1.8	Iconeditor mit Malfunktionen (arp.library nötig)	204
BattleForce 3.01	Kampf mehrerer roboterähnlicher Maschinen	205
Chess 2.0	Schachspiel von Fish 96, grafisch aufgewertet	205
Triple	Verschiedene Grafik- und Sound-Demos	206
MemFlick	Verkettet fragmentierten Speicher	206
PictureGarden	Demo in kompiliertem Basic, nicht lauffähig	206
Hawk	3D-Stereo-Bild eines Adlers	206
Brownian	Demo mit Fractals und Braunscher Bewegung	206
PeX	Demo verschiedener Amiga-Grafikfähigkeiten	206
StereoDemo	3D-Stereo-Animation	206
Coyote	Computerfilm mit Roadrunner-Comicfigur (arced)	207
AsteroidField	Raumschiff fliegt durch Asteroiden-Feld	208
Bowl	Drei Kugeln rotieren in einer Schüssel (zooed)	209
Dps 1.1	Seiten-Previewer für PrintScript-PS-Interpreter	209
NuHand	Frauenhand krallt sich in Tisch, mit Sound	210
Calc 3.0	Simuliert wissenschaftlichen Taschenrechner	210
LabelPrint 1.9	Zum Ausdrucken von Diskettenaufklebern	210
AmigaWave	Amiga-Schriftzug spiegelt sich in Swimming Pool	211
Esperanto	Keymaps und Fonts für Esperanto und Walisisch	211
Image-Ed 1.9	Iconeditor mit Malfunktionen (arp.library nötig)	211
VirusControl 1.3	Virusdetektor mit Bootblock-Archiv und mehr	211
SignFont	Keymap-Datei und Font für Taubstummensprache	211
DiskSalv 1.40	Neue Version, fehlerbereinigt, Zusatzfunktionen	212

© 1990 technicSupport

Alice	Makabere Sculpt3D/Pageflipper-Animation	212
DogsWorld	Hund rennt über rotierende Weltkugel	212
Cucug	Schriftzug kreist um verspiegelte Kugel	213
Icons	Ca. 300 verschiedene, achtfarbige Icons	213
MemDiag 1.1	Testprogramm für den RAM-Speicher	214
MandelVroom 2.0	C-Code für Programm von Fish 215	214
ArcPrep 2.1	Bereitet Verzeichnisse für Arc vor	214
SmartIcon	Programm von Disk 134, mit Source in C	214
RunBack	Arbeitet jetzt mit Null:-Gerät (Disk 188)	214
MandelVroom 2.0	Mandelbrot-Programm	215
C64Emul	Gag: Zeigt Einschaltbild eines C64	216
VirusX 3.20	Neue Version des Virusprotektors	216
Backdrop	Unterlegt Workbench-Hintergrund mit Muster	216
PrtSpool	Spooler und Hilfsprogramm für Druckerausgaben	216
Wanderer 2.2	Sammeln von Diamanten in einem Labyrinth	216
Cloud	Erzeugt fraktale Wolkengebilde	216
Stevie 3.6	Amiga-Version des UNIX-vi-Editors	217
InstallBeep	Ersetzt Bildschirmblitz durch Geräusch	217
SnipIt 1.2	Kopiert Text von einer Console in ein Window	217
SonixPeek	Listet Instrumente aus SONIX-Dateien	217
AntiCBS	Gegen verleumderischen Hacker-Bericht der CBS	217
Echo	Ersatz für den AmigaDOS-Echo-Befehl	217
EdLib 1.1	Zusatz-Library für Manx, Update zu MkLib (183)	218
Worm	Animation mit wählbaren Farbschlangen	218
Scanner 1.0	Bildet C-Code aller Intuition-Strukturen im Ram	218
PcPatch	Patcht PCCopy/PcFormat auf 3 1/2-Zoll	218
Maze 1.2	Grafikdemo, generiert Labyrinth	218
Mandel	Mandelbrot-Programm mit ARexx-Port	218
DeepSky 5.0	Datenbank über 10000 astronomischer Objekte	219
Mv 1.1	Die drei Unix-Befehle Mv, Cp und Rm in einem	219
DNet 2.0	Für Datenaustausch mit UNIX-Systemen	220
DFrame 1.02	Für Bob-Animationen unter DPaintII	221
IFFM2 1.0.0D	Demo eines IFF-Support-Moduls für M2Amiga	221
Steinschlag 1.5	Geschicklichkeitsspiel nach Art von Tetris	221
ANSIEd 1.2.0aD	Demo eines Editors für farbige Texte	221
AllocMaster 1.17	Testet Programmverhalten bei Speicherknappheit	221
BallyII	Umsetzung des Arcade-Spiels Click	221
Mischief	Gag: Simuliert und manipuliert Benutzereingaben	222
MemGauge 1.4	Zeigt Ausnutzung des Speichers an	222
PlPlot 1.00	Lattice-kompatible Bücherei für Plotfunktionen	222
IFFtoSun 1.31	Wandelt IFF-Bild in SUN-Rasterfile-Format	223
Paccer 1.0	PacMan-Spiel mit Soundeffekten und Level-Editor	223
GravSim 1.5	Simuliert Schwerkraftbewegung von 6 Planeten	223
FixDisk 1.0	Versucht, zerstörte Diskette zu reparieren	223
IFF2Sun	Wandelt IFF-Bild in SUN-Rasterfile-Format	223

 © 1990 technicSupport

Csh 3.03a	Verbessert das CLI zu einer Shell	223
PopInfo 3.0	Info-Befehl für die Workbench	223
SetCPU 1.5	Ändert Parameter für 32-Bit-CPU's	223
PetersQuest 1.0	Hüpf-Adventure	224
KickMem 2.0	Bringt AddMem-Kommando im Kickstart unter	224
Xebec	FastFileSystem/Parkfunktion für Xebec-Platten	224
CLImax	Sorgt für randlose CLI-Fenster	224
MoreIsBetter	Aufwertung für das Textanzeigeprogramm More	224
Who	Zeigt Tasks an	224
MyMenu	Erlaubt Zusatzmenüs auf der Workbench	225
AmigaTCP	KA9Q Internet Software Package für DFÜ per Funk	225
Vlt 4.036	VT100- und Tektronix-Emulation mit ARexx-Port	226
RexxArpLib 2.3	ARexx-Interface zur Arp.library (und Intuition)	227
RexxMathLib1.2/3	Höhere mathematische Funktionen für ARexx	227
Midi.Lib	Zusätzliche Library für Midi-Anwendungen	227
PickPacket 1.0	Zum Experimentieren mit DOS-Packets	227
Glib	Soundeditor für TX81Z, DX100, DEP5, DW8000, K-5	228
Xoper 1.3	Leistungsfähiger Systemmonitor	228
JazzBench 0.8	Leistungsfähigerer Ersatz für die Workbench	228
Az 1.4	Brauchbarer Texteditor, unterstützt Umlaute	228
DrawMap	Zum Herstellen von Landkarten	229
Emporos	Strategiespiel um Erwerb eines Landes	229
esuoM	Gag: Mauspointer reagiert spiegelverkehrt	229
LeftyMouse	Vertauscht die Maustasten (für Linkshänder)	229
AlarmingClock	Uhrenprogramm mit Alarmfunktion	229
Sim 4.0	Simulation von Register-Tranfer-Netzen	229
Shuffle	Zum Durchschalten aller Screens mit Ctrl-M	229
AskTask 2/4/89	Taskeditor, der auch Nachrichten schickt	230
Fedup	Dateiorientierter Diskettenmonitor	230
FileIt 1.0	Datenbank in Draco	230
NComm 1.8	Terminalprogramm	230
Quattro 1.0	Geschicklichkeitsspiel nach Art von Tetris	230
PrvHndlr 3	Verhindert 68010-Privilegsverletzung, resetfest	230
Diff	Vergleicht Dateien untereinander	231
File 1.0	Analysiert Dateien auf ihre Funktion	231
NoClick2	Unterdrückt Laufwerksklicken bei A 2000B (Guru)	231
Plot 1.2	Plottet 2D- und 3D-Graphen nach einem Script	231
Sed 1.02	Amiga-Version des UNIX stream editors	231
Dbug	Debugger für C	232
ReSourceDemo3.06	Interaktiver Disassembler	232
Bally III	Umsetzung des Arcade-Spiels Click	232
Brik	Vergleicht Disketteninhalt mit CRC-Checkliste	233
CRCLists	CRC-Checklisten für Fishdisks bis Nr. 231	233
CacheCard 1.00	Kontrolliert MMU-Caching, erg. SetCPU (Fish 223)	233
MuchMore 1.8	Textanzeiger mit weichem Scrolling	234

© 1990 technicSupport

PrintIt 1.0	Druckt IFF-Bilder auf Epson 9-Nadler	234
NetWork 1.0	Gagprogramm	234
XHair 1.0	Mauszeiger wird zu bildschirmgroßem Fadenkreuz	234
WBPic 1.0	Unterlegt Workbench mit einem IFF-Bild	234
KwikBackup 1.0	Schnelles Backup-Programm	234
CalcKey 1.0	Einfacher Pop-Up-Taschenrechner	235
MirrorWars	Spiel mit Laserstrahlen und Spiegeln	235
Ct 2.2	Slideshow und Betrachten von Tomographien	235
AmigaBench	Dhrystone-Benchmark-Programm	236
DiskHandler	Filesystem, liest und schreibt 1.2-Formate	236
Heart3D	Zeigt Drahtmodell eines schlagenden Herzens	236
Ls 3.1	List-Befehl nach UNIX-Art, Update zu Disk 178	236
Proc	Aufruf von DOS-Prozessen ohne LoadSeg()	236
XprZmodem	Library für Datentransfer mit XModem-Protokoll	236
ParOut	Zeigt Kommunikation mit Parallelport (Assembler)	237
CLIPrint	Zeigt Ausgaben an das CLI (Assembler)	237
CType 1.0	Zum Anzeigen von Texten	237
Speed 1.0	Benchmarkprogramm	237
ILBMLib	Library für Lesen und Schreiben von ILBM-Dateien	237
DPlot 1.0	Plottet experimentelle (bin. oder ASCII-) Daten	237
CWDemo 3.1	Ändert die Farben von Screens	238
DMouse 1.20	Mausunterstützung mit PopCli, Hotkeys und mehr	238
LabelPrint 2.5	Zum Ausdrucken von Diskettenaufklebern	238
Steinschlag 1.8	Geschicklichkeitsspiel nach Art von Tetris	238
NGC 1	Virusdetektor, prüft Bootblöcke und Vektoren	238
Pyth 1.1	Zeichnet Baum des Pythagoras	238
JGoodies	Alle Programme dieser Disk mit Source in JForth	239
Textra	Texteditor ohne deutsche Umlaute	239
Evolution	Simuliert Wachstum von Wanzen und Bakterien	239
FFT	Berechnet Fourier-Transformationen	239
Guru	Dekodiert Guru-Fehlernummern zu Klartext	239
H2J	Konvertiert ".h"-Dateien in ".j"-Files	239
HAMmmm2	Grafikdemo mit hypnotischem Effekt	239
HeadClean	Setzt Reinigungsdisketten in Gang	239
JustBeeps	Zeigt Benutzung von Audio- und Timer-Device	239
Mandelbrot	Berechnet Mandelbrot-Grafiken	239
NeuralNets	Simuliert Informationsverarbeitungs-Modell	239
Brunjes	Verschiedene Tools	239
CrossDos	Lesen und Schreiben von MSDOS- u. Atari-Formaten	240
Dis	Libraray und Programm für 68000-Disassembler	240
DM-Maps	Lösungsbilder zum Spiel Dungeon Master	240
RunBack 6	Führt Programm im Hintergrund aus	240
MemLib	Library für Speicherdebugging	240
XprLib	Library für XModem-Protokoll	240
ASDG-RRD	Resetfeste RAM-Disk bis 8MByte	241

© 1990 technicSupport

CBBS 6.1	Mailbox-System für Amateur-Funk	241
Fix68010	Patcht Programme gegen Prioritätsverletzung	241
NoClick 3.4	Unterdrückt Klicken der Laufwerke bei A 2000B	241
Tiles 2.1	Shanghai-Spiel in Modula-2	241
Man 1.2	Verwaltet Dokumentationsdateien	241
FileReq	Dateiauswahlrequester in C	242
JPClock 1.2	Uhr mit Zusatzfunktionen	242
Image-Ed 2.2	Iconeditor mit Malfunktionen (arp.library nötig)	242
PopInfo 3.1	Informiert über Laufwerke, Speicher und Zeit	242
PaletteReq	Ändert die Farbpalette	242
ZeroVirus	Gegen Link- und Bootblockviren	242
FullView	Zeigt Texte mit 80 Spalten	242
Bootblocks	Informationen über Bootblöcke	242
JAR	Grafisches 3D-Spiel	242
CustReq 2	Wie Ask, Antwort erfolgt jedoch über Requester	242
Check4Mem	Informiert per Returncode über Speicherplatz	242
PPrefs	Kompakter Ersatz für die Preferences	242
FragIt 2.0	Zerstückelt den Speicher (zu Testzwecken)	243
WarpUtil 1.11	Komprimieren von Tracks und ganzen Disketten	243
SimGen	Unterlegt Workbench mit IFF-Bild	243
Lines 1.0	Grafikdemo mit einstellbaren Parametern	243
ImageLab 2.2	Zum Verändern bestehender IFF-Bilder	243
Password 1.21p	Erschwert unbefugten Zugriff auf Amiga	243
NoClick 3.5	Unterdrückt Klicken der Laufwerke bei A 2000B	243
Pcopy 2.0	Nützliches Kopierprogramm mit Extras	243
LPE 1.0	Grafikeditor für LaTeX	243
BBChampion 3.1	Archivierung und Analyse von Bootblöcken	244
XColor 1.2	Ändert Farbanzahl- und werte eines Screens	244
SizeChecker 1.2	Vergleicht Dateilängen mit einer Liste	244
Textdisplay 1.52	Zeigt Texte an, mausgesteuert	244
BootIntro 1.2	Kreiert Vorspann mit kurzem Text	244
FMC 1.2	An- und abschaltbares NoFastMem	244
ATOF	Ändert von der WB aus das Fonts:-Verzeichnis	245
BootIntro	Farbenreichere Version des Programms von 244	245
Fenster 1.0	Erlaubt Ändern von Fenster-Parametern	245
Reversi 1.2	Das gleichnamige Brettspiel	245
PathMaster	Dateiauswahlrequester in C	245
Vlt 4.065	VT100- und Tektronix-Emulation mit ARexx-Port	245
Dmake 1.0	Make-Utility	246
ScreenShare 1.2	Library für Screen- und Window-Handling	246
Ty	Textanzeiger mit Maussteuerung auf Less-Basis	246
LabelPrint 2.5b	Zum Ausdrucken von Diskettenaufklebern	246
NoClick 3.6	Unterdrückt Klicken der Laufwerke bei A 2000B	246
NComm 1.8	Terminalprogramm, identisch mit Version von 230	246
AnalogJoystick	Treiberprogramm für analogen Joystick	247

RemoteLogin	Führt auf Signal am ser. Port DOS-Befehl aus	247
XprLib 2.0	Library für XModem-Protokoll	247
AssemTools	140 Makro-Routinen, Library und Beispielprogs	247
AmiGantt 3.0.0	Hilft bei Planung von Projekten	248
Flipper	Reversi-Spiel	248
Regex 1.0	Library für Verarbeitung von Wildcards	248
Maze	Generiert Labyrinthe, re-entrant	248
CLI-Colors	CLI-Fenster erhalten farbige Ränder	248
NetHandler 1.0	Spricht Geräte auf vernetztem (DNet) Amiga an	248
Slicer 1.0	Mandelbrot-Programm	249
Turmite	Simuliert zweidimensionale Turing-Maschine	249
Automata Vier	Zellwachstums-Simulationen	249
ASimplex 1.5	Simplex-Algorithmus für lineare Programme	250
Gravity-Well	Simuliert Bewegung von bis zu 20 Himmelskörpern	250
RPSC 1.1	Programmierbarer Rechner mit UPN-Notation	250
Paranoids	Brettspiel: Flucht aus einer Nervenheilanstalt	250
Debug 2.10	Komfortabler symbolischer Debugger	251
DiskSalv 1.42	Neue Version, fehlerbereinigt	251
DiskSpeed 1.0	Ermittelt Geschwindigkeit von Geräten	251
Monopoly	Umsetzung des bekannten Brettspiels	251
ReadRoute 1.0	Hilft bei Planung einer Reiseroute	251
BasicGames	Drei Basic-Spiele	251
Demon	Zwei Programme, die Labyrinthe zeichnen	251
Mountain	Zeichnet fraktale Gebirgslinie	251
Fish Tank	Fische schwimmen in einem Aquarium	251
Turmites	Zellwchstums-Simulation	251
CrossDOS 3.05b	Konvertieren von MSDOS-, Atari-Formaten	252
TrekTrivia 2.0	Fragequiz zum Thema StarTrek	252
Zerg 1.0	Adventure in der Art von Ultima	252
DumpDiff 1.0	Ermittelt Unterschiede zweier Dateien (HEX)	253
Elements 1.2a	Daten und Periodensystem der Elemente	253
KeyMac 0.1	Belegt Tastatur mit Makros	253
MegaWB 1.2	Vergrößert die Workbench (mit Scrolling)	253
MuchMore 2.5	Textanzeiger mit weichem Scrolling	253
MultiSelect 1.0	Mehrfach-Anwahl von Icons ohne Shift-Taste	253
PowerPacker 2.3b	Komprimiert Datenfiles und Befehle	253
WBShadow 1.0	Versieht alle WB-Objekte mit Schatten	253
With 1.0	Aufruf anderer Befehle mit ARP-Wildcards	253
Etale	Textanzeiger mit griechischem Alphabet	254
MachII 2.6	Maus-Utility mit PopCli, ClickTOFront etc.	254
PatternLib	Library für Wildcard-Verarbeitung	254
SlotCars	Autorenn- und Schießspiel	254
Uedit 2.5b	Flexibler ASCII-Texteditor	254
CyclicSpace 1.42	Zellwachstums-Simulation	255
MakeIcon 1.1	Verwandelt IFF-Bild in Icon und umgekehrt	255

© 1990 technicSupport

MED	Soundeditor wie Soundtracker, erzeugt C-Code	255
RolandD110 1.0	Überträgt Sound-Samples an Roland D-110	255
RolandS220 2.0	Überträgt Sound-Samples an Roland S-220	255
SunMaze	Erzeugt Labyrinth mit Schatten	255
BlitDemons 1.0	Schnelle Zellwachstums-Simulation (Blitter)	256
NameGame	Wortpuzzle-Spiel	256
Stevie 3.7a	Amiga-Version des UNIX-vi-Editors	256
ColorReq	Library für Farbpaletten-Requester	257
FileIO	Dateiauswahlrequester in C und Assembler	257
Vlt 4.226	VT100- und Tektronix-Emulation mit ARexx-Port	257
Backup 2.01	Sichert Inhalt einer Festplatte auf Disketten	258
DMouse 1.20	Mausunterstützung mit PopCli, Hotkeys und mehr	258
STReplay	Library, lädt und spielt Soundtracker-Module	258
SupLib	Library für Matt-Dillons Programme	258
EscapeFromJoviIII	Geschicklichkeitsspiel mit Schwerkraftsimulation	259
Accordion 1.0.1	Demo eines Solitaire-ähnlichen Spiels	260
Calculation 1.0.2	Demo eines Solitaire-ähnlichen Spiels	260
SeaHeaven 1.1	Demo eines Solitaire-ähnlichen Spiels	260
CLib 1.3	C-Library mit ca 140 Funktionen	260
XprZmodem 2.0	Library für Datentransfer mit XModem-Protokoll	261
CopDis 0.0a	Disassembler für Copperlisten	261
ShowBiz	Tierfiguren gestalten eine Theateraufführung	261
Indent 1.1	Formatierer für C-Quellcodes	262
Lotto	Ermittelt Losnummer plus Teilnehmer per Zufall	262
Jumble	Emittelt alle Lösungen eines Anagramms	262
QMouse 1.6	Mausbeschleuniger mit vielen Sonderfunktionen	262
QView 1.1	Sehr schneller Textanzeiger (auch HEX-Modus)	262
WorldDataBank2.0	Datenbank mit geographischen Daten des CIA	262
Sentinel	Badge Killer Siegerbeitrag (3 MB) Teil 1	263
Sentinel	Badge Killer Siegerbeitrag (3 MB) Teil 2	264
VAUX-Killer	Badge Killer Zweitplazierter (3 MB) Teil 1	265
VAUX-Killer	Badge Killer Zweitplazierter (3 MB) Teil 2	266
Mackie 1.3	Bildschirm-Schoner mit Zusatzfunktionen	267
MatLab	Vektor- und Tensor-Rechnung unter ABSoft Fortran	267
DigLib	Grafik-Library für ABSoft Fortran	267
Doctor A	Korbwurf eines Basketballers	268
Klide	Sehenswerte Blitter-Demo	268
OnlyAmiga	Demo, unterstreicht Fähigkeiten des Amiga	268
Calendar	Druckt Kalenderblätter in zehn Standard-Formaten	269
ChessTutor	Schach-Trainingsprogramm in AmigaBasic	269
PropGadget	Proportional-Gadget in C und Assembler	269
RadBoogie	Demo mit Multitasking und Blitter-Programmierung	269
ShowDisk	Zeigt Blöcke einer Datei und ihre Funktionen	269
SpinPointer	Mauszeiger signalisiert Programmaktivität	269
ColumnSet	Arrangiert Textdateien spaltenweise	270

MRBackUp 3.3d	Zum Erstellen von Festplatten-Sicherungskopien	270
TooMuch3D	Raumschiff verläßt den Monitor	270
CPUStandoff	Spaßhafte Demo mit MAC-Beteiligung	271
LaserBoing	AmigaBall als rotierende Spiegelkugel	271
AmigaPunt	Ermittelt Erfolgschancen von Rennpferden	272
Ballon	Preßluft-Flasche füllt Luftballon	272
Spigot	Tropfender Wasserhahn	272
BattleForce	Kampf mehrerer roboterähnlicher Maschinen	273
PennyWise	Girokonto- und Scheckverwaltung	273
HP11 1.1	Simuliert HP11C-Taschenrechner	274
KeyBiz	Gag: Geräusch nagender Mäuse bei Tastendruck	274
LookFor	Sucht nach bestimmten Dateien	274
SlideMaster 0.1	Slideshow-Programm mit ARexx-Port	274
Snap 1.3	Schneidet Text und Grafik aus Screens heraus	274
Xoper 2.0	Leistungsfähiger Systemmonitor	274
Vt100 2.9	Terminalprogramm, VT100-Emulation	275
Blit	Spaßige Visualisierung von Blitter und ECS	276
ClickDOS II 1.10	Erlaubt CLI-Dateioperationen über die Maus	276
DateRequester	Datums- und Zeitrequester für ARexx	276
ARTM 0.9	Vielseitiger Taskeditor	277
Icon 7.5	Programmiersprache Icon	277
LabelPrint 3.0	Zum Ausdrucken von Diskettenaufklebern	277
LPE 1.0	Grafikeditor für LaTeX	277
WBAssign 1.20	Ermöglicht Assign-Zuweisung über Icon	277
WBRes 1.2b	Nutzt residente Programme von der Workbench	277
AmigaTrek	3 StarTrek-Parodien mit Amiga-Bezügen	278
ASpice 5.1	Simuliert elektronische Schaltkreise (ab 1.5 MB)	278
Blocks	Zeigt Blockverteilung von Dateien	278
Free	Zeigt freie Blöcke einer Diskette	278
Ash 1.0	Shell nach UNIX-ksh-Art	279
MouseClock 1.2	Mauszeiger zeigt Speicher und Zeit	279
MRBackUp 3.3e	Zum Erstellen von Festplatten-Sicherungskopien	279
MRPrint 3.4	Erleichertert formatiertes Drucken von Texten	279
Berserker 3.0+	Leistungsstarker Virusprüfer	280
CM 1.0	Simuliert Bewegung von Himmelskörpern	280
Graph 1.0	Plottet mathematische Funktionen	280
Diff 1.10	Vergleicht Dateien miteinander	281
DiffDir 1.1	Vergleicht Verzeichnisinhalte miteinander	281
FarPrint 1.3	C-Debugging ohne Terminal und Debug.lib	281
LoadImage 1.9	Bildbetrachter, zeigt auch Overscan an	281
MRARPFile	Ergänzende I/O-Funktionen zu ARP	281
MRMan 1.0	Man-Variante, verwaltet Dokumentationsdateien	281
PrintHandler 1.1	PRT:-Ersatz für Einzelblätter und Spooling	282
RCS	Verwaltet mehrfache Revisionen von Textdateien	282

 © 1990 technicSupport

Autorenverzeichnis

Werner Görtz, Jahrgang 1960, studierte Anglistik/Romanistik an der Universität Düsseldorf. Während eines Studienaufenthaltes in Berkeley/USA machte er seine erste Bekanntschaft mit dem Phänomen Computer. Über C64 und PC-10 gelangte er Mitte 1986 zum Amiga. Am meisten beschäftigt er sich mit der C-Programmierung, Desktop Publishing und der Public Domain. Zur Zeit ist er bei Computerland in Essen beschäftigt. Von Werner Görtz stammen die Artikel zu `CardFile`, `Chess` und `VC/AnalytiCalc`.

Ralf Leithaus, geboren 1961 in Berlin, studiert an der Freien Universität die Fächer Latein und Physik. 1987 wechselte er vom C64 auf Amiga, den er in erster Linie mit Schriftsatz (TEX) und der Public-Domain auszureizen versucht. Für die Amiga-Welt schreibt er Grundlagenartikel und regelmäßige Berichte über die Public Domain. Als Herausgeber des vorliegenden Buches erfaßte er die neuen Programme für den Listenteil und beschrieb die Programme `AmiCron`, `ARP 1.3`, `AsciiTable`, `AssignDev`, `ConMan`, `DFC2`, `Dme`, `EW`, `FontCon`, `Hexalator`, `Hyperbase`, `IconType`, `KeyClick`, `less`, `MandelVroom`, `MaxWindow`, `Moose`, `MouseZoom`, `Movie`, `NECquik`, `NewZap`, `Paranoid`, `PLAY`, `PointerAnimator`, `PrintPop`, `QView`, `Ren`, `RunBack`, `SetPrefs`, `startle`, `VirusX`, `WBLace`, `WBLander`, `XIcon`, `ZapIcon` sowie die Einrichting der individuellen Workbench und die PD-Story.

Holger Lubitz, geboren 1971 in Hannover, leistet nach bestandenem Abitur momentan seinen Grundwehrdienst ab. Seinen Amiga 1000, den er seit Anfang 1987 besitzt, programmiert er vorwiegend in C und Assembler, ansonsten interessiert er sich auch für Musik und Computerkunst und ist Herausgeber des PD-Diskettenmagazins `AmigaJUICE`. Von ihm stammen die Beschreibungen zu `MachII` und `SafeBoot`.

Stefan Ram, geboren 1962 in Berlin, beschäftigt sich seit 1985 mit Amiga, wobei er sich stark auf die C-Programmierung und den Schriftsatz — in erster Linie mit AmigaTEX — konzentriert. Stefan Ram verfaßte die Beschreibungen zu `Hack` und `TurboBackUp`.

Ottmar Röhrig, geboren 1970 in Frankfurt, kam mit 9 Jahren zum Computer. Seit Anfang 1986 mit Amiga arbeitend, begann er im gleichen Jahr, für verschiedene Computerzeitschriften als freier Mitarbeiter zu schreiben. Seine Hauptinteressen auf dem Amiga liegen bei Anwenderprogrammen, Video/Grafikanwendungen, PD-Software sowie der Programmierung in Modula-2. Auch der Schriftsatz mit TEX ist eins seiner Steckenpferde. Von Ottmar Röhrig stammen die Beschreibungen zu den Programmen `FuncKey`, `Mackie`, `ShoWiz` und `Vdraw`.

Michael Sauer, geboren 1971 in Herne, stieg Juni 1987 von dem PC 128 auf den Amiga um. Er beschäftigt sich hauptsächlich im redaktionellen Bereich und gründete im März 1988 das Diskettenmagazin AMIGA Faszination. Er hat bis dato an drei Büchern mitgeschrieben und interessiert sich auch weiterhin für die Funktion als Autor. Sein weiteres Interesse gilt der Spieleszene, DeskTopVideo, Programmieren von nützlichen Anwendungsprogrammen in BASIC und dem Sammeln von PD-Disketten. Von ihm stammen die Beschreibungen der Gagprogramme.

Axel Schmidt, 1964 in Hamburg geboren, studiert Luft- und Raumfahrttechnik an der TU Berlin und beschäftigt sich seit 1986 mit dem Amiga. Seine Hauptinteressen sind neben dem Schriftsatz mit AmigaTEX die Programmierung in C, Assembler und ARexx. Axel Schmidt hat die Programme `DiskMan` und `Flaschbier` beschrieben.

Katja Thielbeer wurde 1961 in Rastatt geboren und ist als Angestellte beschäftigt. Im November 1987 lernte sie AMIGA kennen und neben Spielen nutzt Sie ihn auch für Public Domain Software. Gerade diesen Bereich hält sie oft für gleichwertig mit kommerzieller Software. Von Katja Thielbeer stammt die Programmbeschreibung zu `StoneAge`.

Karl-Wilhelm Zerulla ist 28 Jahre alt und an der Diplomarbeit für den Abschluß seines Studiums des Wirtschaftsingenieurwesens/Informatik an der Universität (TH) Karlsruhe. Seit Mai 1986 programmiert er auf seinem Amiga — zunächst in C, später nur noch in Modula-2. Karl-Wilhelm Zerulla beschäftigt sich „seit ewigen Zeiten“ mit Public-Domain-Software und interessiert sich dabei für alle Gebiete. Er beschrieb den Psychotest `Freud`.

 © 1990 technicSupport

Register

1-2-3 319
150Windows 161
3D-Wasserpistolen 244
3DBreakOut 266
Abenteuerspiel, grafisches 164
Abspecken einer WB-Disk 51
Adreßdatei 180
Alarm 201
Alderosi, Khalid 354
Alias 289
Alpha-beta-Methode 89
Alphanumerische Tastatur 169
AmiCron 59
Amiga-Zeichensätze 140
AmigaShell 64
AmigaTEX 13
Amulett von Yendor 166
Amulett, das von Yendor 168
AnalytiCalc 318
Animation 13, 251, 278
Anwendungsprogramme 13
Apfelmännchen 214, 221, 225
Arbeitshilfen 13
Arbeitstisch 164
Arbeitsumgebung, Archivieren ..298
ARes 31, 64, 71
ARexx 129
Argumente, mehrfache 34
ARP 25
ARP 1.3 ..... 26, 62
Arp.library 68
ArpInstall 29, 62
ARun 72
ASCII-Codes 74
AsciiTable 74
ASH 26, 64, 102
AShell 30, 32, 64, 71
Assembler 68
Assign 76, 363
AssignDev 76
AtariMeetsAmiga 162
Atkins, Robbie J. 277
Backup 315
Balzer, Christian 61
Basename 71
Batchdatei 54, 356
BCPL 68
Bildanzeiger 301
Bildschirmschoner 209
Blur 161
BootBack 295
Bootblöcke 350
Bootblöcke, Archivieren 295
Bootvorgang 36
BOOZ 328
Boulder Dash 136, 311
Brandt, Willy 246
Breakout-Spiel 266
Brouwer, Andries 175
Brushes 365
Bröseln 160
CAD 13
CardFile 79
CedDemo 49
Chess 88
Clague, Kevin 242
CLI 170
CLI, Editierhilfe 96
CLI-Fenster, volle Größe 243
Cmp 71
Cockatrice, tote 166
Computergrafiken 214
ConMan 26, 31, 37, 70, 96
Copyright 11
Csh 102
Cunningham, Greg 115
Cygnus Ed Professional 117
Dateiverwaltung 44
Dateiverwaltung, menügesteuert 178
Daumenkino 251
Default Tool 79, 358
Delta-Dateien 252
DeluxePaint 365
Demolition 163
Demoversionen 13
Design Grid 281
Desktop Publishing 13
Devices 76
DF2: alias DF1: 76
DFC2 ..... 26, 31, 103
Dia-Shows 301
Die PD-Story 43
Dilbm 251
Dillon, Matthew 133
Director, the 13
Disk-Manager 107

Diskcopy 315
Diskettenkopieren 315
Diskettenmonitor 38, 261
DiskMan 107
Dme 116
DMouse 201
Don Juan 146
DropShadow 161
Druckereinstellungen ändern ... 275
Druckerfonts 140, 142
Druckerkontrolle 258
Duppong, Mike 193
Ed 43, 180, 246, 357, 362
Ed, Patchen des 263
Edit 53
Editierfunktionen, verbesserte ... 98
Editierhilfe 96
Editor 169
Editor, konfigurierbar 116
Effekt, optischer 134
Eingabemaske 181
Eingabeprompt 40
Elch, geschwätziger 244
Emerald Mine 311
Epp, Lyman R. 286
Ernst 171
Eröffnungsbibliothek 90, 93
Esuom 159
Everhart, Glenn C. 334
EW 134
ExplodingWindows 134
ExtrasD 247
FaccII 109
Feedback, akustisches 192
Feldlinien 226
Fenlason, Jay 175
Festplatte, Janus- 78
Fish, Fred 12
Flaschbier 136
Flens 136
Flip 161
Floskeln 26
Floskeltasten 153
Flucht 161
FontCon 140
FontFixer 291
Fonts 140
Formular 45, 180, 187
Frage- und Antwortspiel 86
Framework 320
Freud, Sigmund 145
FuncKey 26, 153, 201
Funktionstastenbelegung .. 153, 201
Gagprogramme 159
Gattegno, A. Caleb 75
Gebührenzähler 201
General Design 189
Geräte, physikalische 76
Geschicklichkeitsspiel 266, 355
Geschäft 171
Goodeve, Pete 191, 364
Grafik 13
Graham, Eric 257
Grenzwert 219
Hack 164
Hawes, William S. 102, 130
Heath, Charlie 62, 294
Heck, Dieter-Thomas 246
Helvetica 140
Henscheid, Eckhard 244
Hexalator 176
Hintergrundprozesse 26
Hintergrundtasks 292
History-Funktion 26, 99
Hodgson, John 265
hot spot 279
Hyperbase 45, 178
Icon-Typ, Ändern 190
IconEd 356
Icons aus IFF-Bildern 365
IconType 190, 357
Ikea-Imperium 244
Imaginärteil 222
Info 79
Info-Tool 191, 356
Ing 162
Interlace einschalten 354
Introduction 160
jh0: alias dh0: 78
Joker 62, 67, 72
Joymouse 159
Julia-Mengen 230
Kalkulation 318
Kalkulationsblatt 318
Karteikasten, elektronischer 79
Keep, Tim 284
KeyClick 31, 192
KillPointer 280
Komplexe Zahlen 214, 220
Komplexe Zahlenebene 222
Koordinatensystem 222
Kopf-über 161
Kopieren von Disketten 315
Kopp, Martin 317
Kramer, Klaus 314
Langeveld, Willy G.J. 300
Lanoux, Mark 297
Lateinvokabeln 80
Laufschrift 309
Lehenbauer, Karl 355
Leivian, Bob 92, 200
Lernhilfe 80
less 194
Level-Editor 267

© 1990 technicSupport

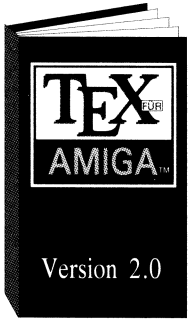
Lindsay, Phillip 78
LoadLib 71
MacGag 161
MachII 26, 31, 201
MacKenzie, Michael 189
Mackie 26, 209
Mah, Anson 158
Makedir 34
Malprogramm 335
Mandelbrot, Benoît B. 220
Mandelbrot-Menge 224, 226
MandelVroom 214
Masken 45, 180
Mausbeschleuniger 250
Mausgeschwindigkeit 250
Mauszeiger, animiert 278
MaxWindow 243
Med 43
Meissen, John 250
Melt 160
MeltILBM 160
Mengel, Marc 189
Merge 278
Merrick, David 147
Midlife-Crisis 147, 150
Monster 164
Moose 244
MooseCtrl 245
Mountlist 35
Mouse 159
MouseZoom 250
Move 71
Movie 251, 307
Muncho 163
Musik 13
Mutitasking 292
MyCli 102
Mätresse des Bischofs 244
NEC P6 141
NECquik 143, 2580
Neeto! 163
Newcli 293
NewShell 30, 102
NewZap 38, 261
Norborg Craig 189
Notepad 140, 291, 362
Oing 162
Open Access 320
Option 16, 174
Orris, Hobie 310
PAL-Erkennung 36
Paranoid 266
Parks, James B. 87
Patchen 37, 259
Payne, Jon 175
Perkins, Bill 244, 249
Persönlichkeitstest 145
Pietzcker, Tim 141
Pilbm 251
Ping 160
Pinsel 365
Pip: 26, 34, 70
PLAY 269, 361
PointerAnimator 278
PopCli 201
Portabilität 68
PrintPop 275
Programmierwerkzeuge 13
Programmstart nach Uhrzeit 59
Prompt 34
Protektor 351
Prt:, Steuersequenzen 188
Quickmouse 159
QView 285
Ray Tracing 252
Read 71
Realteil 221
Regebogen 160
Ren 287
Rename 362
Rexx 129
Riley, Marc 274
Robotroff 162
Rochade 91, 94
Rokicki, Tomas 106, 213
Ross, Scott S. 177
Run 293
RunBack 25, 31, 201, 292
SafeBoot 295
Sampson, Steve R. 61
Sand 162
Say 247
Scat! 161
Schachprogramm 88
Schachprogrammierung 89
Schaeffer, Rick 61
Scherer, Ed 135
Schmelzen 160
Schriftrolle 171
Schwab, Leo L. 310
ScreenBlanker 201
Sculpt 3D 13
Search 67
Seed 231
Seibert, Oliver 78
Sektoren 261
Selbstanalyse 152
Selbstähnlichkeit 230
Serienbrief 45
Set 71
Setmap 158
Setprefs 298
Shareware 12
Shaw, George Bernard 85

Shell 102
Shell-Startup 40
ShoWiz 301
SiliCon 102
Silva, Peter da 355
Slide-Shows, Musikuntermalung 270
SlideShows 301
Software Brewery 61
SONIX 269
Sort 67
Speechd 247
Spreadsheet 318
Sproing 162
Spuren 261
Stack 40
Stanbeck, John 92
Stapelverarbeitungsdatei 357
Startle 309
Startup-Sequence 25
Startvorgang 32
Stempel, Steffen 317
STEmulator 162
Steuersequenzen für prt: 188
Stone-Age 311
SunMouse 204
SunMouse 36
Superbitmap 335
SuperBrickout 266
Symphony 320
System-Configuration 298
TackOn 71
Target 160
Tastatur-Makros 201
Tastencodes 74
Tastenschloß 161
Textbetrachter 285
Textdateien, Durchsuchen von .. 194
Textscrolling 309
Textverarbeitung 13
Thome, Mike 175
Tibbett, Steve 353
Tilt 161
Toebes, John 175
Tool Types 80, 359
Trepal, George 145, 152
Trickfilme 13
Trivial Pursuit 79, 86
TurboBackup 78, 315
Typ, analer 146
Typ, oraler 146
Typ, Phallus- 146
Type 67
TEX 130
Uhr 201
UNIX 169
Unknown command 18
Update-Service 12
Vbrush 348
VC 318
VDraw 335
Vermeulen, Stephen 291, 349
Vgrab 348
Vi 169
ViaCom 162
Viren 295
Virusdetektor 350
VirusX 26, 296, 350
VisiCalc 319
Vokabeltraining 81
Vpad 348
Vpalette 348
WB 1.3 25
WBLace 354
WBLander 355
WBRun 249
Wente, Felix 268
Werner 136
White, J. L. 308
Wildcards 62
Window-Handling 100
Woodland, Kenny 175
Workbench, Aufräumen der 28
WShell 210
Wörterbuchprogramm 85
Xicon 55, 274, 356
Yaboing 162
YaBoing II 160
Yendor, Amulett von 166, 168
Zahlen, komplexe 214, 220
Zahlenfolge 217
Zahlenformate, Umrechnen 176
ZapIcon 357, 365
Zeichensätze 287
Zeichensätze, Auslagern von 287
Zeichensätze, selbstdefinierte ... 141
Zelle 320
Zoing 162
Zwei Jahre Iberer-Forschung 244

© 1990 technicSupport

Schriftsatz für Profis

Ausdrucke in höchster Qualität mit allen Druckertypen



Demo-Version
2 Disks + Broschüre
DM 36,-

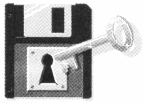
AmigaTeX 2.0 bietet:

* flexiblen und kostengünstigen Schriftsatz
* ca. 100 TeX-Zeichensätze, alle fremdsprachlichen Zeichensätze
* wissenschaftlicher Formelsatz, alle speziellen Zeichensätze
* NEU: Verwendung von Postscript-Schriften in jeder Größe - zusätzlich und gleichzeitig zu den TeX-Schriften
* NEU: Ausdruck von IFF-Rastergraphiken auf allen Druckern
* Vorbetrachtung von Text und Bildern (NEU) mit Preview auf dem Bildschirm
* NEU: Aufruf beliebiger DOS-Kommandos von AmigaTeX aus
* Schnittstelle zu ARexx
* automatischer Umbruch mit Register und Inhaltsverzeichnis
* Kopfzeilen und Fußnoten werden automatisch verwaltet
* Hochwertiger Druck mit allen Druckertypen (z.B. 24-Nadel-Drucker 360 x 360 dpi)
* Ausgabe über Satzbelichter auf Film oder Fotopapier
* deutsche Trenntabelle, deutsches Manual, Update-Service

Import und Localisation: Stefan Ram Programmierung, Berlin

Marketing und Verlag GmbH
Bundesallee 36-37, 1000 Berlin 31, Tel. 030-8621314/5, Fax 030-860496

BAND I



technicSupport

Stefan Ram/Jens Hertwig (Hrsg.)

DAS GROSSE AMIGA PUBLIC DOMAIN BUCH, Band I

352 Seiten, div. Abbildungen
ISBN 3-926847-01-8

Das Standardwerk zur PD-Software für den Amiga mit über 40 deutschsprachigen Anleitungen zu den beliebtesten Programmen.

Aus dem Inhalt:
- Anwenderprogramme
- Grafiksoftware unter der PD
- Arbeits- und CLI-Hilfen
- Programmierhilfen.

Dokumentiert sind die PD-Reihen FISH, PANORAMA und FAUG.
Alle Programmanleitungen sind nach einem gleichen Schema aufgebaut und von den besten PD-Experten leicht verständlich beschrieben.

DM 49,-

BAND II

technicSupport

Ralf Leithaus/J. Hertwig (Hrsg.)

DAS ZWEITE AMIGA PUBLIC DOMAIN BUCH, Band II

384 Seiten, div. Abbildungen
ISBN 3926847-05-0

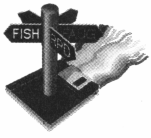
Die Fortsetzung der Buchreihe mit deutschsprachigen Anleitungen zu weiteren 46 wichtigen PD Programmen und neuen Listen zur Software.

Aus dem Inhalt:
- Spiele
- Anwenderprogramme
- Grafik, Animation, Musik
- Utilities und CLI-Hilfen
- Programmiersprachen/Utilities

Alle Programme sind leicht verständlich und sofort bedienbar dargestellt. Dokumentation der Reihen FISH und RPD mit Kurzbeschreibungen im Anhang.

DM 49,-

BAND III



technicSupport

Ralf Leithaus/J. Hertwig (Hrsg.)

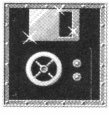
DAS DRITTE AMIGA PUBLIC DOMAIN BUCH, Band III

416 Seiten, div. Abbildungen
ISBN 3-926847-06-9

Mehr als 90 neue PD-Programme sind in Band III der Reihe zur Public Domain Software beschrieben. Ausführliche deutschsprachige Anleitungen garantieren jedem Leser, daß er alle Programme sofort bedienen kann. Fortgesetzt wird auch die "PD-Story" mit ausführlichen Beispielen zur Anwendung der frei kopierbaren Amiga Software.
Dokumentiert sind die Reihen FISH, FAUG, ACS, TBAG, PANORAMA, AUSTRIA. FISH-Disks sind bis Disk-Nr. 172 beschrieben.

DM 49,-

BAND IV



technicSupport

Ralf Leithaus/J. Hertwig (Hrsg.)

DAS GOLDENE AMIGA PUBLIC DOMAIN BUCH, Band IV

512 Seiten, div. Abbildungen
ISBN 3-926847-08-5

Die Krönung der Buchreihe zur PD-Software für Amiga mit mehr als 60 neuen Programmanleitungen in deutscher Sprache.

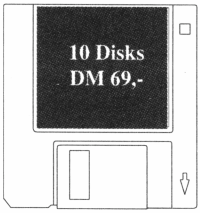
Dieses Buch ist der Abschluß unserer Reihe zur PD-Software und bringt nochmal das beste und aktuellste der beliebten PUBLIC DOMAIN Software.

DM 69,-

DIE PD-BIBLIOTHEK

Band I-IV, 1600 Seiten

DM 216,-



Alle 44 Programme zu Band I

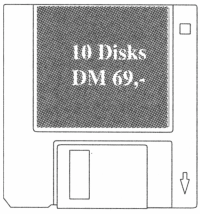
Hier sind alle 44 in Band I beschriebenen Programme auf 10 Disketten. Darunter leistungsfähige Software für: Datenübertragung, Editieren, Grafik, für wissenschaftliche Anwendungen, Slide-Shows, Demo zu Amiga-TEX, Formatieren, resetfeste Ram-Disk, Fonts, Printhilfen, record und replay, IFF-Hilfen, Retten von zerstörten Disketten, Speicherübersichten, Menübildung, Anzeigen von Textdateien, ANSI-Terminal, für Textbearbeitung, Ray-Tracing, das Katalogisieren von Dateien, Datenaustausch, Farbeinstellungen, Komprimieren von Programmen, Umleitung von Druckerausgaben auf Diskette, Gerätetreiber, Spiele und vieles andere mehr.

technicSupport

Alle 46 Programme zu Band II

Hier sind alle 46 in Band II beschriebenen Programme auf 11 Disketten. Darunter anwenderorientierte Software für folgende Bereiche: Kontoverwaltung, Textverarbeitung, Bootblock-Archiv, Kopierprograme, Ray-Tracing, die Grafikbearbeitung, 2D-Konstruktion, Dateiverwaltung, Disketteneditor, Belegung der Funktionstasten, Umrechnen von Zahlenformaten, animierte Icons, Laufschriften, Editieren, Datensicherheit, Fonts, Virusdetektor, Dateiarchivierung, Modula-2-Compiler, 68000-Disassembler, SONIX-Player, Eingabehilfen für das CLI, neue ARP-DOS-Befehle, Animation von IFF-Bildern, Zeichensatz-Editor, Spiele, Persönlichkeitstest und vieles andere mehr.

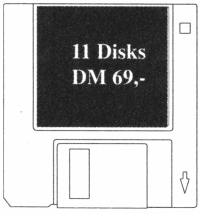
technicSupport



Mehr als 90 Programme zu Band III

Hier sind mehr als 90 der in Band III beschriebenen PD-Programme auf 10 Disketten. Darunter Software für: Dateiverwaltung, Biorhythmus, fraktale Grafik, Landschaftmaler, Textmanipulation, Disketten-Vorspänne, Bearbeiten von ASCII-Dateien, Etikettendruck, Druckhilfen, Editierhilfen, Backup-Programm. Verschlüsseln von Dateien, Metacomco-kompatibler Assembler, Übersetzungsprogramm von C nach Assembler, Modellbahnsteuerung, Schallplatten-Verwaltung, Terminkalender, Mauszeiger-Bibliothek, leistungsfähiges Malprogramm, Spiele, Bildschirmgags, Disk-Manager, Retten von Dateien und vieles andere mehr.

technicSupport



Mehr als 60 Programme zu Band IV

Hier sind mehr als 60 der in Band IV beschriebenen PD-Programme auf 11 Disketten. Darunter Software für: Diskettenmonitor zum Patchen, Bootblock-Retter, Manipulieren von Icons, Etiketten-Druck, Tabellenkalkulation, schnelleres Formatieren, bessere DOS-Befehle, riesiger WB-Screen, Taschenrechner mit Plot-Funktionen, Werner-Spiel, Malprogramm und vieles andere mehr.

DAS GOLD-PAKET

PD-Bibliothek (Bd I-IV , ca. 1.600 S.)
plus 42 Disks / 240 Programme
zum Gesamtpreis von nur

DM 298,-

technicSupport

Offizielle Commodore Amiga Produktübersicht

Komplette Produktübersicht
352 Seiten, farbige Abbildungen,
Software, Hardware und Literatur
ISBN 3-926847-28-X DM 20.-

technicSupport
Marketing und Verlag

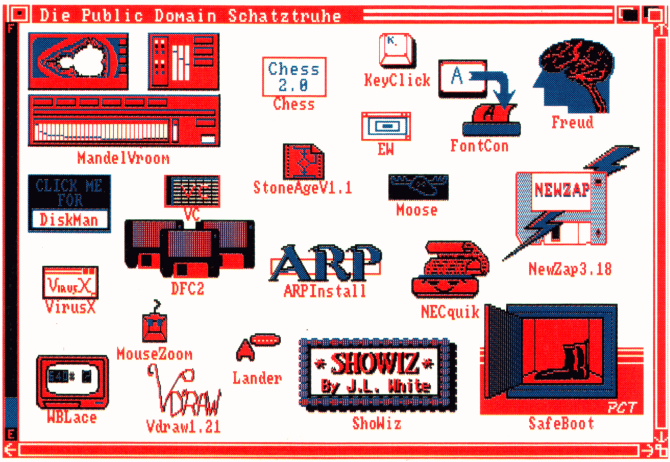
technicSupport's
Wegweiser durch freie AMIGA-Software

Immer mehr AMIGA-Anwender nutzen die Qualität, Vielseitigkeit und das große Angebot der preisgünstigen PD-Programme, um sich ihre Software-Wünsche zu erfüllen.
Tausende von PD-Programmen decken auch die Anwendungsbereiche ab, die von kommerzieller Software nicht oder nur ungenügend berücksichtigt werden.

Dieses komplett mit AMIGA erstellte Buch beschreibt die wichtigsten und erfolgreichsten PD-Programme für den AMIGA. Ausführliche deutschsprachige Anleitungen helfen beim Einstieg in die faszinierende Welt der frei kopierbaren Software und geben auch dem fortgeschrittenen Anwender wertvolle Tips und Hinweise.

Aus dem Inhalt:
- Elektronischer Dateikasten
- Mal- und Zeichenprogramme
- Professioneller Texteditor
- Mandelbrot-Grafiken
- Werner Spiel, PD-Story
- Anti-Virus-Programm
- Tabellenkalkulation
- Animationsprogramme
- Individuelle Workbench
- Übersicht über alle Fishdisks

Ralf Leithaus, geboren 1961 in Berlin, ist als Herausgeber der vierbändigen PD-Bibliothek und des Lernkurses "AmigaDOS für Anwender" bekannt. In Zusammenarbeit mit einem eingespielten Team amigabegeisterter Autoren entstand dieses Buch, das eine neue Reihe preiswerter Taschenbücher für den Amiga begründet.



Highlights aus der Public Domain mit ausführlichen deutschen Anleitungen

ISBN 3-926847-22-0

ÖS 140,- sFr 17,- **DM 19,80**

This was brought to you

from the archives of

http://retro-commodore.eu